Qualys API V2

User Guide

Version 8.5

August 10, 2015

CONTINUOUS SECURITY

Copyright 2007-2015 by Qualys, Inc. All Rights Reserved.

Qualys, the Qualys logo and QualysGuard are registered trademarks of Qualys, Inc. All other trademarks are the property of their respective owners.

Qualys, Inc.
1600 Bridge Parkway
Redwood Shores, CA 94065
1 (650) 801 6100

CONTENTS

Preface

Chapter 1 Welcome

Get Started 9
Get API Notifications 9
Introducing Qualys API V2 10
Qualys User Account 10
API Conventions 11
API Limits 14

Chapter 2 Using the API V2 Architecture

Authentication Using the V2 APIs 18
Using the API V2 Session Resource 20
Session Login 24
Session Logout 26

Chapter 3 Scan API

VM Scans 30
- VM Scan List 30
- Launch a VM Scan 34
- Launch a VM Scan on EC2 Hosts 36
- Take Actions on VM Scans 38

PC and SCAP Scans 41
- PC Scan List 41
- SCAP Scan List 43
- Launch a PC Scan 45
- Launch a PC Scan on EC2 hosts 47
- Take Actions on PC Scans 49

Scan Schedules 52
Scan List Parameters 58
Scan Parameters 61
Scan Schedule Parameters 65
Scanner Appliances 67
- Scanner Appliance List 67
- Manage VLANs and Static Routes 70

Virtual Scanner Appliances 75
KnowledgeBase API 79
Share PCI Scan 83

Chapter 4 Report API

About Report Share 90
Report List 91
Launch Report 94
Launch Scorecard 110
Cancel Running Report 118
Download Saved Report 120
Delete Saved Report 123

Chapter 5 Asset API

IP List 126
Host List 129
Host List Detection 136
Excluded Hosts List 157
Excluded Hosts Change History 160
Manage Excluded Hosts 164
Purge Hosts 168
Virtual Host List 172
Take Actions on Virtual Hosts 173
Restricted IPs List 175
Add IPs 181
Update IPs 183
Asset Group API 185

Chapter 6 IPv6 Asset API

API Support for IPv6 Asset Management and Scanning 194
View IPv6 Mapping Records 203
Add IPv6 Mapping Records 205
Remove IPv6 Mapping Records 207

Chapter 7 Compliance API

Compliance Control List 210
Compliance Policy List 216
Compliance Policy - Import and Export 222
Compliance Policy - Merge 226
Compliance Policy - Manage Asset Groups 232
Compliance Posture Information 235
Control Criticality 241
SCAP Cyberscope Report 241
SCAP ARF Report 246
SCAP Policy List 247

Chapter 8 Scan Authentication API

About Authenticated Scanning 254
User Permissions and Authentication Records 256
Authentication Record List 257
Authentication Record List by Type 260
Unix / Cisco IOS Record 264
Windows Record 275
Oracle Record 283
Oracle Listener Record 292
SNMP Record 297
IBM DB2 Record 306
VMware Record 312
Application Server Records 317
HTTP Record 321
MySQL Record 325
MS SQL Record (compliance scans only) 331
Authentication Vault 338

Chapter 9 Network API

Network List 350
Create Network 351
Update Network 352
Assign Scanner Appliance to Network 353

Appendix A Scan and Report XML

Simple Return 356
Batch Return 358
Scan List Output 360
SCAP Scan List Output 363
Scheduled Scan List Output 366
Vulnerability Scan Results 373
Compliance Scan Results 374
PCI Scan Share Status Output 381
Scanner Appliance List Output 383
Scanner Appliance Create Output 392
KnowledgeBase Output 394
Report List 407
Network List 410

Appendix B Asset XML

IP List Output 414
Host List Output 416
Host List Detection Output 421
Excluded Hosts List Output 426
Excluded Hosts Change History Output 428

Virtual Host List Output 431
IPv6 Mapping Records List Output 433
Restricted IPs List Output 435
Asset Group List Output 437

Appendix C Compliance Data XML

Compliance Control List Output 444
Compliance Policy List Output 449
Compliance Policy Export Output 455
Compliance Posture Information Output 461
Compliance Policy Report 474
Compliance Authentication Report 487
Compliance Scorecard Report 492
SCAP Policy List Output 500

Appendix D Scan Authentication XML

Authentication Record List Output 506
Authentication Record List by Type Output 510
Authentication Record Return 523
Authentication Vault List Output 525
Authentication Vault View Output 528

Appendix E Scorecard Report XML

Asset Group Vulnerability Report 532
Ignored Vulnerabilities Report 535
Most Prevalent Vulnerabilities Report 539
Most Vulnerable Hosts Report 543
Patch Report 547

Appendix F Error Codes / Descriptions

Index

Preface

Using the Qualys API, third parties can integrate their own applications with Qualys cloud security and compliance solutions using an extensible XML interface. The API functions described in this guide are available to customers with Qualys Vulnerability Management (VM) and Policy Compliance (PC).

About Qualys

Qualys, Inc. (NASDAQ: QLYS) is a pioneer and leading provider of cloud security and compliance solutions with over 7,700 customers in more than 100 countries, including a majority of each of the Forbes Global 100 and Fortune 100.The Qualys Cloud Platform and integrated suite of solutions help organizations simplify security operations and lower the cost of compliance by delivering critical security intelligence on demand and automating the full spectrum of auditing, compliance and protection for IT systems and web applications. Founded in 1999, Qualys has established strategic partnerships with leading managed service providers and consulting organizations including Accenture, Accuvant, BT, Cognizant Technology Solutions, Dell SecureWorks, Fujitsu, HCL Comnet, InfoSys, NTT, Tata Communications, Verizon and Wipro. The company is also a founding member of the Cloud Security Alliance (CSA).

For more information, please visit www.qualys.com.

Contact Qualys Support

Qualys is committed to providing you with the most thorough support. Through online documentation, telephone help, and direct email support, Qualys ensures that your questions will be answered in the fastest time possible. We support you 7 days a week, 24 hours a day. Access support information at www.qualys.com/support/.

CHAPTER 1

Welcome

The Qualys API allows third parties to integrate their own applications with Qualys cloud security and compliance solutions using an extensible XML interface. The API functions described in this guide are available to customers with Qualys Vulnerability Management (VM) and Policy Compliance (PC).

Get Started

This chapter gives you an introduction to the Qualys API v2 and how to make requests using this API. We'll discuss API conventions and best practices to get you up and running quickly.

Please note that Qualys API V1 functions are still available for use. For many integrations, users will want to use both the API V1 and V2 functions. Refer to the Qualys API User Guide for documentation on the V1 APIs.

Get API Notifications

We recommend you join our Community and subscribe to our API notifications so you'll get email notifications telling you about important upcoming API enhancements and changes.

From our Community

Join our Community

Subscribe to API Notifications (select Receive email notifications)

Introducing Qualys API V2

The Qualys API V2 is based on a newer API architecture that provides many features and benefits to Qualys API customers. The API V2 architecture is optimized for performance and security, and will be the basis for future Qualys API functionality.

The API V2 architecture provides API versioning and requires an API session login/logout sequence. See Chapter 2, Using the API V2 Architecture.

The Qualys API V2 includes:

- Scan API
- Report API
- Asset API
- IPv6 Asset API
- Compliance API
- Scan Authentication API

Qualys User Account

Authentication with valid Qualys user account credentials is required for making Qualys API requests to the Qualys API servers. These servers are hosted at the Qualys platform, also referred to as the Security Operations Center (SOC), where your account is located. When calling the V2 API functions, users have the option to choose session based authentication using login and logout operations, or basic HTTP authentication using uuencoded credentials passed with each request. See “Authentication Using the V2 APIs” in Chapter 2 for information.

If you need assistance with obtaining a Qualys account, please contact your Qualys account representative.

Users with a Qualys user account may access the API functions. When a subscription has multiple users, all users with any user role (except Contact) can use the Qualys API. Each user’s permissions correspond to their assigned user role.

Qualys user accounts enabled with Two Factor Authentication cannot be used with the Qualys API.

API Conventions

Before using Qualys API functions, please review the API conventions below.

URL to the Qualys API Server

Qualys maintains multiple Qualys platforms. The Qualys API server URL that you should use for API requests depends on the platform where your account is located.

Account Location	API Server URL
Qualys US Platform 1	https://qualysapi.qualys.com
Qualys US Platform 2	https://qualysapi.qg2.apps.qualys.com
Qualys EU Platform	https://qualysapi.qualys.eu
Qualys Private Cloud Platform	https://qualysapi.<customer_base_url>

The Qualys API documentation and sample code use the API server URL for Qualys US Platform 1. If your account is located on another platform, please replace this URL with the appropriate server URL for your account.

Authentication

Authentication with valid Qualys account credentials is required for making Qualys API requests. When calling the V2 API functions, users have the option to choose between session based authentication (using login and logout operations) and basic HTTP authentication (identical method used for the V1 API functions).

See "Authentication Using the V2 APIs" in Chapter 2 for information.

GET and POST Methods

Qualys API V2 functions allow API users to submit parameters (name=value pairs) using the GET and/or POST method. There are known limits for the amount of data that can be sent using the GET method, and these limits are dependent on the toolkit used. Please refer to the individual descriptions of the API function calls to learn about the supported methods for each function.

Parameters in URLs

API parameters, as documented in this user guide, should be specified one time for each URL. In the case where the same parameter is specified multiple times in a single URL, the last parameter takes effect and the previous instances are silently ignored.

Date Format in API Results

The Qualys API has adopted a date/time format to provide consistency and interoperability of the Qualys API with third-party applications. The date format follows standards published in RFC 3339 and ISO 8601, and applies throughout the Qualys API.

The date format is:

yyyy-mm-dd**T**hh-mm-ss**Z**

This represents a UTC value (GMT time zone).

URL Encoding in API Code

You must URL encode variables when using the Qualys API. This is standard practice for HTTP communications. If your application passes special characters, like the single quote (‘), parentheses, and symbols, they must be URL encoded.

For example, the pound (#) character cannot be used as an input parameter in URLs. If “#” is specified, the Qualys API returns an error. To specify the “#” character in a URL you must enter the encoded value “%23”. The “#” character is considered by browsers and other Internet tools as a separator between the URL and the results page, so whatever follows an un-encoded “#” character is not passed to the Qualys API server and returns an error.

UTF-8 Encoding

The Qualys API uses UTF-8 encoding. The encoding is specified in the XML output header as shown below.

```
<?xml version="1.0" encoding="UTF-8" ?>
```

URL Elements are Case Sensitive

URL elements are case sensitive. The sample URL below will retrieve a previously saved scan report that has the reference code “scan/987659876.19876”. The parameter name “ref” is defined in lower-case characters. This URL will return the specified scan report:

```
https://qualysapi.qualys.com/msp/scan_report.php?
ref=scan/987659876.19876
```

The sample URL below is incorrect and will not return the specified scan report because the parameter name “Ref” appears in mixed-case characters:

```
https://qualysapi.qualys.com/msp/scan_report.php?
Ref=scan/987659876.19876
```

Decoding XML Reports

There are a number of ways to parse an XML file. Select the method which is most appropriate for your application and its users.

Qualys publishes DTDs for each report on its Web site. For example, the URL to the scan report can be found at the URL shown below:

```
https://qualysapi.qualys.com/scan-1.dtd
```

The URLs to current report DTDs are included with the function descriptions in this document. There is a generic report returned by a few functions.

Occasionally Qualys updates the report DTDs. It is recommended that you request the most recent DTDs from the Qualys platform to decode your reports. The URLs to the report DTDs are included in this user guide.

Detailed information about each XML report is provided in the appendices at the end of this document. For each XML report a recent report DTD and the report's XML elements and attributes (XPaths) are described in detail.

Some parts of the XML report may contain HTML tags or other special characters (such as accented letters). Therefore, many elements contain CDATA sections, which allow HTML tags to be included in the report. “High” ASCII and other non-printable characters are escaped using question marks.

API Limits

The service enforces limits on the API calls subscription users can make. The limits apply to the use of all APIs, except "session" V2 API (session login/logout).

Important! All API controls are applied on a subscription basis.

Concurrency and Rate Limits

Default settings are provided and these may be customized per subscription by Support.

Concurrency Limit per Subscription (per API). The maximum number of concurrent API call instances allowed within the subscription for each API. Default is 2.

Rate Limit per Subscription (per API). The maximum number of API calls allowed per day (or a customized period, in seconds) within the subscription for each API. The rate limit is defined by the rate limit count and rate limit period. The default rate limit count is 300. The default rate limit period is 86400 seconds (24 hours).

The service checks the concurrency limit and rate limit each time an API request is received. In a case where an API call is received and the service determines a limit has been exceeded, the API call is blocked and an error is returned (the concurrency limit error takes precedence).

Please see the document *Qualys API Limits* for complete information.

API Usage

Your subscription's API usage and quota information is exposed in the HTTP response headers generated by Qualys APIs (all APIs except "session" V2 API).

HTTP Response Headers

The HTTP response headers generated by Qualys APIs are described below.

Note: The HTTP status code "OK" (example: "HTTP/1.1 200 OK") is returned in the header for normal (not blocked) API calls. The HTTP status code "Conflict" (example: "HTTP/1.1 409 Conflict") is returned for API calls that were blocked.

Header	Description
X-RateLimit-Limit	Maximum number of API calls allowed in any given time period of <number-seconds> seconds, where <number-seconds> is the value of X-RateLimit-Window-Sec.
X-RateLimit-Window-Sec	Time period (in seconds) during which up to <number-limit> API calls are allowed, where <number-limit> is the value of X-RateLimit-Limit.

Header	Description
X-RateLimit-Remaining	Number of API calls you can make right now before reaching the rate limit <number-limit> in the last <number-seconds> seconds.
X-RateLimit-ToWait-Sec	The wait period (in seconds) before you can make the next API call without being blocked by the rate limiting rule.
X-Concurrency-Limit-Limit	Number of API calls you are allowed to run concurrently.
X-Concurrency-Limit-Running	Number of API calls that are running right now (including the one identified in the current HTTP response header).

Sample HTTP Response Headers

Sample 1: Normal API call (API call not blocked)

Returned from API call using HTTP authentication.

```
HTTP/1.1 200 OK
Date: Fri, 22 Apr 2011 00:13:18 GMT
Server: qweb
X-RateLimit-Limit: 15
X-RateLimit-Window-Sec: 360
X-Concurrency-Limit-Limit: 3
X-Concurrency-Limit-Running: 1
X-RateLimit-ToWait-Sec: 0
X-RateLimit-Remaining: 4
Transfer-Encoding: chunked
Content-Type: application/xml
```

Sample 2: API Call Blocked (Rate Limit exceeded)

Returned from API call using HTTP authentication.

```
HTTP/1.1 409 Conflict
Date: Fri, 22 Apr 2011 00:13:18 GMT
Server: qweb
X-RateLimit-Limit: 15
X-RateLimit-Window-Sec: 360
X-Concurrency-Limit-Limit: 3
X-Concurrency-Limit-Running: 1
X-RateLimit-ToWait-Sec: 181
X-RateLimit-Remaining: 0
Transfer-Encoding: chunked
Content-Type: application/xml
```

Sample 3: API V2 Call Blocked (Concurrency Limit exceeded)

Returned from API V2 call using API V2 session authentication.

```
HTTP/1.1 409 Conflict
Date: Fri, 22 Apr 2011 00:13:18 GMT
Server: qweb
Expires: Mon, 24 Oct 1970 07:30:00 GMT
Cache-Control: post-check=0,pre-check=0
Pragma: no-cache
X-RateLimit-Limit: 15
X-RateLimit-Window-Sec: 360
X-Concurrency-Limit-Limit: 3
X-Concurrency-Limit-Running: 3
Transfer-Encoding: chunked
Content-Type: application/xml
```

Note: In the case where the concurrency limit has been reached, no information about rate limits will appear in the HTTP headers.

Activity Log within User Interface

The Activity Log within the Qualys user interface shows details about user activities actions taken using the user interface and the API.

To view the Activity Log, log into your Qualys account. Go to VM > Users and click the Activity Log tab. Select Filters > Recent API Calls. Uou'll see the API Processes list showing the API calls subject to the API limits (all APIs except "session" V2 API) made by subscription users and/or updated by the service in the past week.

Tip: You can search the processes list to find API processes. You can search by process state (Queued, Running, Expired, Finished and/or Blocked), by submitted date and by last updated date. You can search for API processes that were blocked due to exceeding the API rate limit and/or the API concurrency limit.

CHAPTER 2

Using the API V2 Architecture

The latest Qualys API Suites are based on a new API architecture that provides new features and benefits to Qualys API customers. The new API architecture is optimized for performance and security, and will be the basis for future Qualys APIs.

When using APIs based on the V2 API architecture, you'll notice these features:

- API Versioning — An API version number is incorporated in V2 API URLs. This will allow us to introduce significant API updates from one release to the next in the future, while allowing customers to choose when to migrate to the latest version.
- API Login/Logout Sequence — Authentication with valid Qualys user account credentials is required for making Qualys API requests to the Qualys API servers. When calling the V2 API functions, users have the option to choose between session based authentication (using login and logout operations) and basic HTTP authentication (identical method used for the V1 API functions).

This chapter provides important information about the API V2 architecture and the session login/logout functions. These topics are covered:

- Authentication Using the V2 APIs
- Using the API V2 Session Resource
- Session Login
- Session Logout

Authentication Using the V2 APIs

Authentication with valid Qualys account credentials is required for making Qualys API requests to the Qualys API servers. When calling the V2 APIs, users have the option to choose between session based authentication (using login and logout operations) and basic HTTP authentication (identical method used for the V1 API functions). Specifying the "X-Requested-With" parameter is required for all V2 API requests regardless of the authentication method used.

The code examples that follow assume that the user's account is located on US Platform 1 with the base URL https://qualysapi.qualys.com. As a reminder, if your account is located on another platform please replace this with the base URL for your platform.

Required Header Parameter

The following header parameter must be included in all API v2 calls using basic HTTP authentication and session based authentication:

```
"X-Requested-With: <user description, like a user agent>"
```

Specifying the required "X-Requested-With" parameter helps to protect Qualys API users from cross-site request forgery (CSRF) attacks.

See sample API requests below for examples using Curl.

Using Basic HTTP Authentication

Using this method, Qualys account credentials are transmitted using the "Basic Authentication Scheme" over HTTPS for each API call. For information, see the "Basic Authentication Scheme" section of RFC #2617:

```
http://www.faqs.org/rfcs/rfc2617.html
```

The exact method of implementing authentication will vary according to which programming language is used.

Sample API Request (Curl)

A sample asset/host API request (Curl) using basic HTTP authentication:

```
curl -H "X-Requested-With: Curl Sample" -u "acme_ab12:passwd"
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/?action=list"
```

Using Session Based Authentication

Using this method, the user makes a sequence of API V2 requests as follows:

Step 1: Make session login request

Use the Qualys API **session** resource to make a login request. Upon success, the request returns a session ID in the Set-Cookie HTTP header:

```
curl -H "X-Requested-With: Curl Sample" -D headers
-d "action=login&username=acme_ab12&password=passwd"
"https://qualysapi.qualys.com/api/2.0/fo/session/"
```

Step 2: Make resource requests

Use the V2 API resources to make V2 API requests, as described in this user guide, and include the session ID in the cookie header for each request.

You'll notice the session cookie (QualysSession) was extracted from the "headers" file contents returned from the session login API call (Step 1 above):

```
curl -H "X-Requested-With: Curl Sample"
-b "QualysSession=71e6cda2a35d2cd404cddaf305ea0208; path=/api;
secure" -d "action=list"
"https://qualysapi.qualys.com/api/2.0/fo/report/"
```

Step 3: Make session logout request

Once logged in to Qualys you can make multiple API requests. Use the Qualys API **session** resource to logout of the current session. Logging out of the session closes the open session and ensures secure, ongoing access to your account. Access may be denied if a user makes too many session login requests without closing sessions properly:

```
curl -H "X-Requested-With: Curl Sample"
-b "QualysSession=10b8eb6d4553b4d1ecb860c2b3c247d4; path=/api;
secure" -d "action=logout"
"https://qualysapi.qualys.com/api/2.0/fo/session/"
```

Using the API V2 Session Resource

Sessions created using the Qualys API via the **`session`** resource are equivalent in every way to sessions created by users logging into the Qualys user interface. Too many open sessions, whether created via the API and/or via user interface login, will lock out new session login attempts from both interfaces (user and API).

The request URL has several elements. The following elements appear in every request URL based on the API V2 architecture.

URL element	Description
qualysapi.qualys.com:443	FQDN of the Qualys API server and option port (443 if specified).
api	Qualys Application component name.
2.0	Qualys API version number.
fo	Qualys interface component name.
session | scan | report	Qualys API resource name. In the sample session login URL above, the resource "session" is specified for the session login request. For a reporting request, the resource "report" is used.
action={value}	Qualys API resource-specific action. In the sample session login URL above, the action is "login".

Session Login Request

The session login request includes the Qualys user login credentials, the request URL, and the location where the HTTP response headers will be saved.

The sample API call below saves the HTTP headers in a local file named "headers":

```
curl -H "X-Requested-With: Curl Sample" -D headers
-d "action=login&username=acme_ab12&password=passwd"
"https://qualysapi.qualys.com/api/2.0/fo/session/"
```

If you do not wish to store this information in the "headers" file, you can save the HTTP header in a cookie as shown below:

```
curl -H "X-Requested-With: Curl Sample" -c cookie.txt
-d "action=login&username=acme_ab12&password=passwd"
"https://qualysapi.qualys.com/api/2.0/fo/session/"
```

Upon success, the sample Qualys API call returns an XML response with the message "Logged in" and the Qualys API session ID in the Set-Cookie HTTP header. See "HTTP Response Headers" for further information.

Resource Requests

When session based authentication is used, the session cookie returned in the XML response from the session login request must be included in the cookie header of subsequent API requests. Multiple API requests can be made using the same session cookie.

The resource request includes the Qualys user login credentials, the Qualys API session ID, the request URL, and the location where the HTTP response headers are saved.

The sample API request below is used to request a list of reports in the user's Report Share storage space. You'll notice the session cookie (QualysSession) was extracted from the "headers" file contents returned from the session login API call.

```
curl -H "X-Requested-With: Curl Sample"
-d "action=list"
-b "QualysSession=71e6cda2a35d2cd404cddaf305ea0208; path=/api;
secure" "https://qualysapi.qualys.com/api/2.0/fo/report/"
```

If you saved the HTTP response headers (from the session login request) in a cookie file, make an API request to obtain the cookie from the cookie file as shown below:

```
curl -H "X-Requested-With: Curl Sample"
-d "action=list"
-b "cookie.txt"
"https://qualysapi.qualys.com/api/2.0/fo/report/"
```

Upon success, the sample report list API call returns an XML response listing the reports in the user's Report Share. In progress and completed reports are included.

HTTP Response Headers

These API requests return HTTP response headers: session login requests, session logout requests, and fetch (download) report requests. These requests provide information to the third party application about the XML output.

Sample XML output showing HTML response headers returned from a session logout request is shown below:

```
HTTP/1.1 200 OK
Date: Wed, 20 Jun 2007 16:21:03 GMT
Server: qweb/3.3h
Set-Cookie: QualysSession=71e6cda2a35d2cd404cddaf305ea0208;
path=/api; secure
Expires: Mon, 24 Oct 1970 07:30:00 GMT
Cache-Control: post-check=0,pre-check=0
Pragma: no-cache
Connection: close
Transfer-Encoding: chunked
Content-Type: text/xml
```

Sample XML output showing HTML response headers returned from a fetch (download) report request, where the report format is HTML, is shown below:

```
HTTP/1.1 200 OK
Date: Wed, 20 Jun 2007 16:36:42 GMT
Server: qweb/3.3h
Expires: Mon, 24 Oct 1970 07:30:00 GMT
Cache-Control: post-check=0,pre-check=0
Pragma: no-cache
Content-Disposition: attachment;
filename=scan_report__1182357402.zip
Content-length: 98280
Connection: close
Content-Type: application/zip
```

Expires HTTP Header. For the Expires header, Qualys complies with RFC #2109 and sets the Expires date to an old date (a date long in the past). Currently Qualys sets the Expires date to "Mon, 24 Oct 1970 07:30:00 GMT". Note that Qualys cookie expiration is managed on the server side, and Qualys does not rely on clients to drop their expired cookies.

Session Logout Request

A sample session logout request (POST method) is shown below. Upon success, the sample Qualys API call returns an XML response with the message "Logged out".

```
curl -H "X-Requested-With: Curl Sample"
-d "action=logout"
-b "QualysSession=71e6cda2a35d2cd404cddaf305ea0208; path=/api;
secure" "https://qualysapi.qualys.com/api/2.0/fo/session/"
```

See "Session Logout" later in this chapter for further information.

Session Timeout

Every Qualys user account has a session timeout setting. This setting is configurable at the subscription level by Manager users in the Qualys user interface (go to Users > Setup > Security). For a new subscription, this is set to 60 minutes.

The session timeout applies to sessions started using the user interface and sessions started using the Qualys APIs, including APIs based on the new API architecture.

When you launch a scan or report (using Report Share), the task is launched in the background, and processing does not timeout until the task has completed.

Session Login

The **`/api/2.0/fo/session`** resource is used to make a request for session login. Parameters are described below.

Parameter	Description
action=login	(Required) A flag used to make a session login request.
username	(Required) The user name (login) of a Qualys user account.
password	(Required) The password of a Qualys user account.
echo_request={0|1}	(Optional) Specifies whether to echo the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.

A session login request is used to authenticate to the Qualys API and receive a Qualys API session ID, which must be included in the cookie header of subsequent API resource requests. Thus, a session login request is required before making API calls to the Scan API or the Report Share API.

The POST access method must be used to make a session login request.

A sample session login request (POST method) is shown below. Upon success, the sample Qualys API call returns an XML response with the message "Logged in" and the Qualys API session ID as shown.

```
curl -H "X-Requested-With: Curl Sample" -D headers.4
-d "action=login&username=acme_ab12&password=passwd"
"https://qualysapi.qualys.com/api/2.0/fo/session/"

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE GENERIC SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">

<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2007-06-20T16:21:04Z</DATETIME>
    <TEXT>Logged in</TEXT>
  </RESPONSE>
</SIMPLE_RETURN>
```

```
cat headers.4

HTTP/1.1 200 OK
Date: Wed, 20 Jun 2007 16:21:03 GMT
Server: qweb/3.3h
Set-Cookie: QualysSession=71e6cda2a35d2cd404cddaf305ea0208;
path=/api; secure
Expires: Mon, 24 Oct 1970 07:30:00 GMT
Cache-Control: post-check=0,pre-check=0
Pragma: no-cache
Connection: close
Transfer-Encoding: chunked
Content-Type: text/xml
```

Session Logout

The **/api/2.0/fo/session** resource is used to make a request for session logout. Parameters are described below.

Parameter	Description
action=logout	(Required) A flag used to make a session logout request.
echo_request={0|1}	(Optional) Specifies whether to echo the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.

When you're done making V2 API resource requests, the third party application must make a session logout request. This results in closing the session ID for the user's account, preventing future API requests from running.

The POST access method must be used to make a session logout request.

A sample session logout request (POST method) is shown below. Upon success, the sample Qualys API call returns an XML response with the message "Logged out" as shown.

```
curl -H "X-Requested-With: Curl Sample"
-d "action=logout"
-b "QualysSession=71e6cda2a35d2cd404cddaf305ea0208; path=/api;
secure" "https://qualysapi.qualys.com/api/2.0/fo/session/"

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE GENERIC SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2007-06-20T21:50:37Z</DATETIME>
    <TEXT>Logged out</TEXT>
  </RESPONSE>
</SIMPLE_RETURN>

cat headers.18

HTTP/1.1 200 OK
Date: Wed, 20 Jun 2007 21:50:36 GMT
Server: qweb/3.3h
Expires: Mon, 24 Oct 1970 07:30:00 GMT
```

```
Cache-Control: post-check=0,pre-check=0
Pragma: no-cache
Set-Cookie: QualysSession=71e6cda2a35d2cd404cddaf305ea0208;
expires=Wed, 13-Jun-2007 21:50:37 GMT; path=/fo
Connection: close
Transfer-Encoding: chunked
Content-Type: text/xml
```

CHAPTER 3

Scan API

The Scan API V2 provides a suite of API fun ctions for managing vulnerability scans and compliance scans across the enterprise.

This chapter describes how to use Scan API functions to manage scans in your user account. These topics are covered:

- VM Scans
- PC and SCAP Scans
- Scan Schedules
- Scan List Parameters
- Scan Parameters
- Scan Schedule Parameters
- Scanner Appliances
- Virtual Scanner Appliances
- KnowledgeBase API
- Share PCI Scan

VM Scans

The VM Scan API v2 (**`/api/2.0/fo/scan/`**) is used to obtain a list of vulnerability scans in your account and to take actions on them like cancel, pause, resume, and fetch (download) finished results. Authentication is required for each API request. See Chapter 2, "Authentication Using the V2 APIs."

Express Lite: This API is available to Express Lite users.

Permissions

User Role	Permissions
Manager	Manage scans on all IPs in the subscription.
Unit Manager	View scans with targets containing IPs in the user's business unit. Download scan results when the target includes at least one IP in the user's business unit.
Scanner	View scans with targets containing IPs in the user's account. Download scan results when the target includes at least one IP in the user's account.
Reader	View scans with targets containing IPs in the user's account. Download scan results when the target includes at least one IP in the user's account.
Auditor	No permission to manage a scan.

VM Scan List

The VM Scan List API v2 (**`/api/2.0/fo/scan/?action=list`**) lists vulnerability scans in the user's account. By default the XML output lists scans launched in the past 30 days. The GET or POST access method may be used to make a scan list request. Authentication is required to make a request. See Chapter 2, "Authentication Using the V2 APIs."

Benefits

Using the V2 API **`scan`** function for listing scans in your account provides these benefits over using the V1 API scan_report_list.php function:

- Ability to make a single API request to view all scans in your scan history list, including running and completed scans, paused and resumed scans.
- More efficient performance especially when downloading a large amount of scan data from your Qualys account.

- Supports the most scalable and efficient method for integration of host detection data and vulnerability data in the KnowledgeBase when used in conjunction with the scan fetch API (/scan/?action=fetch) and the knowledgebase API (/api/2.0/fo/knowledge_base/vuln/?action=list).
- More input parameters to filter out scans from the scan list output, so you don't have to retrieve and view the entire scan history list as it appears in your account.
- Numerous ways to precisely filter scans based on the when scans were launched. By default, a scan list request selects scans launched within the past 30 days. You may choose a longer window (for example within the past 60 days) or a custom window with specific start and end times (for example starting 2010-02-25T22:42:00Z and ending 2010-02-27T08:00:00Z).

Parameters

The input parameters for requesting a VM scan list are shown below. Please see "Scan List Parameters" for complete details.

Type	Parameter List
Request	action=list (required), echo_request
Scan List Filters	scan_ref, state, processed, type, target, user_login, launched_after_datetime, launched_before_datetime
Show Information	show_ags, show_op, show_status, show_last

Sample Requests

Sample 1. The sample scan list API call (GET method) below returns all scans in the user account.

```
curl -H "X-Requested-With: Curl Sample"
-b "QualysSession=71e6cda2a35d2cd404cddaf305ea0208; path=/api;
secure" "https://qualysapi.qualys.com/api/2.0/fo/scan/
?action=list&echo_request=1&show_ags=1&show_op=1"

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SCAN_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/scan/scan_list_output.
dtd">
<SCAN_LIST_OUTPUT>
  <REQUEST>
    <DATETIME>2007-09-25T12:28:29Z</DATETIME>
    <USER_LOGIN>acme_ab</USER_LOGIN>
    <RESOURCE>https://qualysapi.qualys.com/api/2.0/fo/scan/
```

```
    </RESOURCE>
    <PARAM_LIST>
      <PARAM>
        <KEY>action</KEY>
        <VALUE>list</VALUE>
      </PARAM>
      <PARAM>
        <KEY>echo_request</KEY>
        <VALUE>1</VALUE>
      </PARAM>
      <PARAM>
        <KEY>show_ags</KEY>
        <VALUE>1</VALUE>
      </PARAM>
      <PARAM>
        <KEY>show_op</KEY>
        <VALUE>1</VALUE>
      </PARAM>
    </PARAM_LIST>
  </REQUEST>
  <RESPONSE>
    <DATETIME>2007-09-25T12:28:29Z</DATETIME>
    <SCAN_LIST>
      <SCAN>
        <REF>scan/1187117392.587</REF>
        <TYPE>On-Demand</TYPE>
        <TITLE><![CDATA[Web Servers 09/25]]></TITLE>
        <USER_LOGIN>acme_ab</USER_LOGIN>
        <LAUNCH_DATETIME>2014-09-25T08:10:43Z</LAUNCH_DATETIME>
        <DURATION>00:05:16</DURATION>
        <PROCESSED>1</PROCESSED>
        <STATUS>
          <STATE>Finished</STATE>
        </STATUS>
        <TARGET><![CDATA[10.10.10.10-10.10.10.113]]></TARGET>
        <OPTION_PROFILE>
            <TITLE><![CDATA[Initial Options]]></TITLE>
            <DEFAULT_FLAG>1</DEFAULT_FLAG>
        </OPTION_PROFILE>
      </SCAN>
      <SCAN>
```

```
          <REF>scan/1169604974.6553</REF>
          <TYPE>Scheduled</TYPE>
          <TITLE><![CDATA[Web Servers]]></TITLE>
          <USER_LOGIN>acme_sb3</USER_LOGIN>
          <LAUNCH_DATETIME>2014-09-24T15:40:02Z</LAUNCH_DATETIME>
          <DURATION>00:05:16</DURATION>
          <PROCESSED>0</PROCESSED>
          <STATUS>
            <STATE>Finished</STATE>
          </STATUS>
          <TARGET><![CDATA[10.10.10.10-10.10.10.113]]></TARGET>
          <OPTION_PROFILE>
              <TITLE><![CDATA[Initial Options]]></TITLE>
              <DEFAULT_FLAG>1</DEFAULT_FLAG>
          </OPTION_PROFILE>
        </SCAN>
      </SCAN_LIST>
    </RESPONSE>
  </SCAN_LIST_OUTPUT>
  ...
```

Sample 2. The sample scan list API call (GET method) below returns all running scans that were launched by the user with the login ID "acme_ab":

```
curl -H "X-Requested-With: Curl Sample"
-b "QualysSession=71e6cda2a35d2cd404cddaf305ea0208; path=/api;
secure" "https://qualysapi.qualys.com/api/2.0/fo/scan/
?action=list&state=Running&user_login=acme_ab"
```

Sample 3. The sample scan list API call (GET method) below returns all scheduled scans that were launched after September 3, 2007.

```
curl -H "X-Requested-With: Curl Sample"
-b "QualysSession=71e6cda2a35d2cd404cddaf305ea0208; path=/api;
secure" "https://qualysapi.qualys.com/api/2.0/fo/scan/
?action=list&type=Scheduled&launched_after_datetime=2007-09-03"
```

XML Output

The scan list output uses the scan list output DTD (scan_list_output.dtd). This DTD can be found at the following URL (when your account is located on US Platform 1):

```
https://qualysapi.qualys.com/api/2.0/fo/scan/scan_list_output.dtd
```

Launch a VM Scan

The VM Scan API v2 (**`/api/2.0/fo/scan/?action=launch`**) is used to launch vulnerability scans in the user's account.

A major benefit of using the new Launch Scan API v2 is that it is asynchronous. When you make a request to launch a scan using this API, the service will return a scan reference ID right away and the call will quit without waiting for the complete scan results. We recommend using this new Launch Scan API v2 instead of the Launch Scan API v1 (/msp/scan.php) for this reason.

Parameters

The input parameters for launching a VM scan are shown below. Please see "Scan Parameters" for complete details.

Type	Parameter List
Request	action=launch (required), echo_request, runtime_http_header
Scan Title	scan_title
Option Profile	option_id or option_title
Scanner Appliance	iscanner_id or iscanner_name
Asset IPs/Groups	ip, asset_group_ids, asset_groups, exclude_ip_per_scan, default_scanner, scanners_in_ag
Asset Tags	target_from=tags, use_ip_nt_range_tags, tag_include_selector, tag_exclude_selector, tag_set_by, tag_set_exclude, tag_set_include
Network	ip_network_id (when the Network Support feature is enabled)

Sample Requests

Launch Vulnerability Scan API v2 Request (IP Address):

This request launches a vulnerability scan on the IP address 10.10.10.10 using the scanner appliance "scanner1".

```
curl -H "X-Requested-With: Curl" -u "USERNAME:PASSWORD" -X "POST"
-d
"action=launch&scan_title=My+Vulnerability+Scan&ip=10.10.10.10&opt
ion_id=43165&iscanner_name=scanner1"
"https://qualysapi.qualys.com/api/2.0/fo/scan/" > outputfile.txt
```

XML Output:

```
cat outputfile.txt

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2013-01-15T21:32:40Z</DATETIME>
    <TEXT>New vm scan launched</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>136992</VALUE>
      </ITEM>
      <ITEM>
        <KEY>REFERENCE</KEY>
        <VALUE>scan/1358285558.36992</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

Launch Vulnerability Scan API v2 Request (Asset Tags):

This request launches a vulnerability scan on hosts with the asset tag Windows using the scanner appliance "scanner1".

```
curl -H "X-Requested-With: Curl" -u "USERNAME:PASSWORD" -X "POST" -d
"action=launch&scan_title=My+Vulnerability+Scan&target_from=tags&t
ag_set_by=name&tag_set_include=Windows&option_id=43165&iscanner_na
me=scanner1" "https://qualysapi.qualys.com/api/2.0/fo/scan/" >
file.txt
```

Launch a VM Scan on EC2 Hosts

The VM Scan API v2 (**`/api/2.0/fo/scan/?action=launch`**) is also used to launch vulnerability scans on your Amazon EC2 hosts (in your Amazon Web Services account).

The Amazon *EC2 Scan* workflow using Qualys is pre-authorized by AWS. Want to learn more? Check out our Help Center for Amazon Web Services at the Qualys Community.

A few things to consider...

- EC2 Scanning must be enabled for your Qualys account.
- Only a Manager user can launch EC2 scans.
- Before scanning you'll need to complete some set up steps. See Getting Started with Amazon EC2 Pre-Authorized Scanning at the Qualys Community.

Parameters

The input parameters for launching an EC2 scan are shown below. Please see "Scan Parameters" for complete details.

Type	Parameter List
Request	action=launch (required), echo_request
Scan Title	scan_title
EC2 environment	connector_name (required), ec2_endpoint (required)
Option Profile	option_id or option_title
Scanner Appliance	iscanner_id or iscanner_name
Target Hosts	target_from=tags (required) Use tags to select the EC2 hosts you want to scan.
	use_ip_nt_range_tags=0 The default setting is "0". Important - This cannot be set to "1" for EC2 scanning.
	These tag parameters are used to select tags: tag_set_include={tag1,tag2,...} (required) tag_set_exclude={tag1,tag2,...} (optional) tag_include_selector={**any** | all} (default in bold) tag_exclude_selector={**any** | all} (default in bold) tag_set_by={**id** | name} (default in bold)

Sample Requests

Launch EC2 Vulnerability Scan API v2 Request:

This request launches an EC2 vulnerability scan using the connector "EC2_Connector" on assets that match tags with IDs 1558997 and 1559222.

```
curl -H "X-Requested-With: Curl" -u "USERNAME:PASSWORD" -X "POST"
-d
"action=launch&scan_title=My+EC2+Scan&connector_name=EC2_Connector
&ec2_endpoint=us-east-1&target_from=tags&use_ip_nt_range_tags=0
&tag_include_selector=any&tag_set_by=id&tag_set_include=1558997,15
59222&option_id=43165&iscanner_name=EC2-1"
"https://qualysapi.qualys.com/api/2.0/fo/scan/" > outputfile.txt
```

XML Output:

```
cat outputfile.txt

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2014-02-25T21:32:40Z</DATETIME>
    <TEXT>New vm scan launched</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>136992</VALUE>
      </ITEM>
      <ITEM>
        <KEY>REFERENCE</KEY>
        <VALUE>scan/1358285558.36992</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

Take Actions on VM Scans

The VM Scan API v2 (**`/api/2.0/fo/scan/?action={action}`**) allows users to take actions on vulnerability scans in their account, like cancel, pause, resume, delete and fetch completed scan results.

Parameter	Description
action={action}	(Required) One action required for the request: cancel - Stop a scan in progress (POST method) pause - Stop a scan in progress and change status to "Paused" (POST method) resume - Restart a scan that has been paused (POST method) delete - Delete a scan in your account (POST method) fetch - Download scan results for a scan with status of "Finished", "Canceled", "Paused" or "Error" (GET or POST method)
echo_request={**0**|1}	(Optional) Specify 1 to echo the input parameters in the XML output. When unspecified, parameters are not listed in the XML output.
scan_ref={value}	(Required) The scan reference for a vulnerability scan. This will have the format: scan/nnnnnnnnnn.nnnnn

Parameters

The parameters used to manage VM scans are described below.

Parameter	Description
action={action}	(Required) An action for the request: cancel - stop a scan in progress, "Running" or "Paused" pause - stop a scan in progress and change status to "Paused" resume - restart a scan that has been paused fetch - download scan results for a scan with the status "Finished", "Canceled", "Paused" or "Error".
echo_request={**0**|1}	(Optional) Specifies whether to echo the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
scan_ref={value}	(Required) Specifies a scan reference. A scan reference has the format "scan/987659876.19876".
ips={value}	(Optional for a fetch request) Show only certain IP addresses/ranges in the scan results. One or more IPs/ranges may be specified. A range entry is specified using a hyphen (for example, 10.10.10.1-10.10.10.20). Multiple entries are comma separated.

Parameter	Description
mode={**brief**|extended}	(Optional for a fetch request) The verbosity of the scan results details. One verbosity mode may be specified: brief (the default) or extended. The brief output includes this information: IP address, DNS hostname, NetBIOS hostname, QID and scan test results if applicable. The extended output includes the brief output plus this extended information: protocol, port, an SSL flag ("yes" is returned when SSL was used for the detection, "no" is returned when SSL was not used), and FQDN if applicable.
output_format={**csv**|json}	(Optional for a fetch request) The output format of the vulnerability scan results. One output format may be specified: csv (the default), or json for JavaScript Object Notation. See Appendix A for information on the output formats.

Sample Requests

Sample 1: Cancel a VM Scan. A sample API call to cancel a scan (POST method) is shown below.

```
curl -H "X-Requested-With: Curl Sample"
-d "action=cancel&scan_ref=234234234.12345"
-b "QualysSession=71e6cda2a35d2cd404cddaf305ea0208; path=/api;
secure" "https://qualysapi.qualys.com/api/2.0/fo/scan/"
```

Sample 2: Pause a VM Scan. A sample API call to pause a scan (POST method) is shown below.

```
curl -H "X-Requested-With: Curl Sample"
-d "action=pause&scan_ref=234234234.12345"
-b "QualysSession=71e6cda2a35d2cd404cddaf305ea0208; path=/api;
secure" "https://qualysapi.qualys.com/api/2.0/fo/scan/"
```

Sample 3: Resume a VM Scan. A sample API call to resume a scan (POST method) is shown below.

```
curl -H "X-Requested-With: Curl Sample"
-d "action=resume&scan_ref=234234234.12345"
-b "QualysSession=71e6cda2a35d2cd404cddaf305ea0208; path=/api;
secure" "https://qualysapi.qualys.com/api/2.0/fo/scan/"
```

XML Output

A request to cancel, pause or resume a scan returns XML output using the DTD "simple_return.dtd", which can be found at the following URL (where <qualysapi.qualys.com> identifies the Qualys API server URL where your account is located):

```
https://qualysapi.qualys.com/api/2.0/simple_return.dtd
```

The DTD for the simple return XML output is provided in Appendix A.

PC and SCAP Scans

The PC Scan API v2 (**`/api/2.0/fo/scan/compliance/`**) is used to launch compliance scans, get a list of compliance scans in your account, and manage them.

The SCAP Scan API v2 (**`/api/2.0/fo/scan/scap/`**) is used to get a list of SCAP scans in your account.

Authentication is required for each API request. See Chapter 2, "Authentication Using the V2 APIs."

Permissions

To use this API, these options must be enabled in the user's subscription: Policy Compliance (PC) module and New Scanner Services. Role-based user permissions are described below.

User Role	Permissions
Manager	Manage compliance scans on all compliance IPs in the subscription.
Unit Manager	When the "Manage compliance" permission is enabled in the user's account settings: 1) ability to launch, list and fetch compliance scans on IPs in the user's business unit, 2) ability to take actions on scans launched by users in the same business unit (cancel, pause, resume and delete).
Scanner	When the "Manage compliance" permission is enabled in the user's account settings: 1) ability to launch, list and fetch compliance scans on IPs in the user's account, 2) ability to take actions on scans that the user owns (cancel, pause, resume and delete).
Reader	No permissions to manage compliance scans.
Auditor	No permissions to manage compliance scans.

PC Scan List

The PC Scan List API v2 (**`/api/2.0/fo/scan/compliance/`** with **`action=list`**) gives you a list of compliance scans in your account. By default the XML output lists scans launched in the past 30 days.

The input parameters for requesting a PC scan list are below. Please see "Scan List Parameters" for complete details.

Type	Parameter List
Request	action=list (required), echo_request
Scan List Filters	scan_id (compliance scan ID), scan_ref, state, processed, type, target, user_login, launched_after_datetime, launched_before_datetime
Show Information	show_ags, show_op, show_status, show_last

API Request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d
"action=list&state=Finished&scan_ref=compliance/1344842952.1340"
"https://qualysapi.qualys.com/api/2.0/fo/scan/compliance/"
```

XML Output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SCAN_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/scan/scan_list_output.dtd
">
<SCAN_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2012-09-12T07:28:46Z</DATETIME>
    <SCAN_LIST>
      <SCAN>
        <ID>3332486</ID>
        <REF>compliance/1344842952.1340</REF>
        <TYPE>Scheduled</TYPE>
        <TITLE><![CDATA[MY PC Scan]]></TITLE>
        <USER_LOGIN>USERNAME</USER_LOGIN>
        <LAUNCH_DATETIME>2012-08-13T07:30:09Z</LAUNCH_DATETIME>
        <DURATION>00:06:29</DURATION>
        <PROCESSED>1</PROCESSED>
        <STATUS>
          <STATE>Finished</STATE>
        </STATUS>
        <TARGET><![CDATA[10.10.25.50]]></TARGET>
      </SCAN>
    </SCAN_LIST>
  </RESPONSE>
</SCAN_LIST_OUTPUT>
```

DTD:

The scan list output DTD can be found at the following URL (when your account is located on US Platform 1):

```
https://qualysapi.qualys.com/api/2.0/fo/scan/scan_list_output.dtd
```

SCAP Scan List

The SCAP Scan List API v2 (**`/api/2.0/fo/scan/scap/`** with **`action=list`**) gives you a list of SCAP scans in your account. By default the XML output lists scans launched in the past 30 days.

The input parameters for requesting a SCAP scan list are below. Please see "Scan List Parameters" for complete details.

Type	Parameter List
Request	action=list (required), echo_request
Scan List Filters	scan_id (compliance scan ID), scan_ref, state, type, target, user_login, launched_after_datetime, launched_before_datetime
Show Information	show_ags, show_op, show_status, show_last

API request 1: all SCAP scans

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -d "action=list"
"https://qualysapi.qualys.com/api/2.0/fo/scan/scap/"
```

API request 2: SCAP scan by reference number

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -d
"action=list&scan_ref=qscap/1402642816.80342"
"https://qualysapi.qualys.com/api/2.0/fo/scan/scap/"
```

API request 3: On Demand SCAP scans only

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -d
"action=list&type=On-Demand"
"https://qualysapi.qualys.com/api/2.0/fo/scan/scap/"
```

XML Output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SCAN_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/scan/scap/qscap_scan_list_output
.dtd">
<SCAN_LIST_OUTPUT>
```

```
  <RESPONSE>
    <DATETIME>2014-06-13T22:56:19Z</DATETIME>
    <SCAN_LIST>
      <SCAN>
        <ID>6980366</ID>
        <REF>qscap/1402694682.80366</REF>
        <TYPE>On-Demand</TYPE>
        <TITLE><![CDATA[<IMG
SRC="http://www.google.com/images/logos/ps_logo2.png">]]></TITLE>
        <POLICY>
          <ID>39298</ID>
          <TITLE><![CDATA[Policy A]]></TITLE>
        </POLICY>
        <USER_LOGIN>acme_ab</USER_LOGIN>
        <LAUNCH_DATETIME>2014-06-13T21:24:42Z</LAUNCH_DATETIME>
        <STATUS>
          <STATE>Finished</STATE>
        </STATUS>
        <TARGET><![CDATA[10.10.30.244, 10.10.34.222]]></TARGET>
...
    </SCAN_LIST>
  </RESPONSE>
</SCAN_LIST_OUTPUT>
```

DTD:

The SCAP scan list output DTD can be found at the following URL (when your account is located on US Platform 1):

```
https://qualysapi.qualys.com/api/2.0/fo/scan/qscap_scan_list_output.dtd
```

Launch a PC Scan

The PC Scan API v2 (**`/api/2.0/fo/scan/compliance/?action=launch`**) is used to launch compliance scans. The POST method is required.

Parameters

The input parameters for launching a PC scan are shown below. Please see "Scan Parameters" for complete details.

Type	Parameter List
Request	action=launch (required), echo_request, runtime_http_header
Scan Title	scan_title
Option Profile	option_id or option_title
Scanner Appliance	iscanner_id or iscanner_name
Asset IPs/Groups	ip, asset_group_ids, asset_groups, exclude_ip_per_scan, default_scanner, scanners_in_ag
Asset Tags	target_from=tags, use_ip_nt_range_tags, tag_include_selector, tag_exclude_selector, tag_set_by, tag_set_exclude, tag_set_include

Sample API Request to Launch a PC Scan

A sample API request to launch a compliance scan is below. You'll notice the XML output uses the simple return DTD (simple_return.dtd).

API Request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d
"action=launch&ip=10.10.25.52&iscanner_name=iscan_er5&option_title=In
itial+PC+Options&echo_request=1"
"https://qualysapi.qualys.com/api/2.0/fo/scan/compliance/" >
apiOutputScan.txt
```

XML Output:

```
cat apiOutputScan.txt

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <REQUEST>
    <DATETIME>2012-09-11T05:28:51Z</DATETIME>
```

```
    <USER_LOGIN>USERNAME</USER_LOGIN>
<RESOURCE>https://qualysapi.qualys.com/api/2.0/fo/scan/compliance/
</RESOURCE>
    <PARAM_LIST>
      <PARAM>
        <KEY>action</KEY>
        <VALUE>launch</VALUE>
      </PARAM>
      <PARAM>
        <KEY>ip</KEY>
        <VALUE>10.10.25.52</VALUE>
      </PARAM>
      <PARAM>
        <KEY>option_title</KEY>
        <VALUE>Initial PC Options</VALUE>
      </PARAM>
      <PARAM>
        <KEY>iscanner_name</KEY>
        <VALUE>iscan_er5</VALUE>
      </PARAM>
      <PARAM>
        <KEY>echo_request</KEY>
        <VALUE>1</VALUE>
      </PARAM>
    </PARAM_LIST>
  </REQUEST>
  <RESPONSE>
    <DATETIME>2012-09-11T05:28:57Z</DATETIME>
    <TEXT>New compliance scan launched</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>3335943</VALUE>
      </ITEM>
      <ITEM>
        <KEY>REFERENCE</KEY>
        <VALUE>compliance/1347341337.35943</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

Launch a PC Scan on EC2 hosts

The PC Scan API v2 (**`/api/2.0/fo/scan/compliance/?action=launch`**) is also used to launch PC scans on your Amazon EC2 hosts (in your Amazon Web Services account).

The Amazon *EC2 Scan* workflow using Qualys is pre-authorized by AWS. Want to learn more? Check out our Help Center for Amazon Web Services at the Qualys Community.

A few things to consider...

- EC2 Scanning must be enabled for your Qualys account.
- Only a Manager user can launch EC2 scans.
- Before scanning you'll need to complete some set up steps. See Getting Started with Amazon EC2 Pre-Authorized Scanning at the Qualys Community.

Parameters

The input parameters for launching an EC2 scan are shown below. Please see "Scan Parameters" for complete details.

Type	Parameter List
Request	action=launch (required), echo_request
Scan Title	scan_title
EC2 environment	connector_name (required), ec2_endpoint (required)
Option Profile	option_id or option_title
Scanner Appliance	iscanner_id or iscanner_name
Target Hosts	target_from=tags (required) Use tags to select the EC2 hosts you want to scan.
	use_ip_nt_range_tags=0 The default setting is "0". Important - This cannot be set to "1" for EC2 scanning.
	These tag parameters are used to select tags: tag_set_include={tag1,tag2,...} (required) tag_set_exclude={tag1,tag2,...} (optional) tag_include_selector={**`any`** | all} (default in bold) tag_exclude_selector={**`any`** | all} (default in bold) tag_set_by={**`id`** | name} (default in bold)

Sample Requests

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST"
-d
"action=launch&scan_title=My+EC2+Scan+via+API&connector_name=EC2-
Connector-Lab&ec2_endpoint=us-east-
1&target_from=tags&tag_include_selector=any&tag_set_by=id&tag_set_
include=270325&option_id=61769&iscanner_name=my-ec2-scanner"
"https://qualysapi.qualys.com/api/2.0/fo/scan/compliance/"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <REQUEST>
    <DATETIME>2014-06-24T10:10:51Z</DATETIME>
    <USER_LOGIN>USERNAME</USER_LOGIN>
<RESOURCE>https://qualysapi.qualys.com/api/2.0/fo/scan/compliance/
</RESOURCE>
  </REQUEST>
  <RESPONSE>
    <DATETIME>2014-06-24T10:10:57Z</DATETIME>
    <TEXT>New compliance scan launched</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>2222345</VALUE>
      </ITEM>
      <ITEM>
        <KEY>REFERENCE</KEY>
        <VALUE>compliance/1347771234.36444</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

Take Actions on PC Scans

The PC Scan API v2 (**`/api/2.0/fo/scan/compliance/?action={action}`**) allows users to take actions on compliance scans in their account, like cancel, pause, resume, delete and fetch completed scan results.

Parameter	Description
action={action}	(Required) One action required for the request: cancel - Stop a scan in progress (POST method) pause - Stop a scan in progress and change status to "Paused" (POST method) resume - Restart a scan that has been paused (POST method) delete - Delete a scan in your account (POST method) fetch - Download scan results for a scan with status of "Finished", "Canceled", "Paused" or "Error" (GET or POST method)
echo_request={**0** | 1}	(Optional) Specify 1 to echo the input parameters in the XML output. When unspecified, parameters are not listed in the XML output.
scan_ref={value}	(Required) The scan reference for a compliance scan. This will have the format: compliance/nnnnnnnnnn.nnnnn

Sample API Request to Fetch PC Scan Results

A sample API request to fetch (download) PC scan results is below. The PC scan must have the status Finished in order to dowload the scan results. You'll notice the XML output uses the compliance scan result output DTD.

API Request:

```
curl -u USERNAME:PASSWORD -H "X-Requested-With: Curl"
"https://qualysapi.qualys.com/api/2.0/fo/scan/compliance/?
action=fetch&scan_ref=compliance/1347709693.37303" >
apiOutputScanFetch.txt
```

XML Output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE COMPLIANCE_SCAN_RESULT_OUTPUT SYSTEM

"https://qualysapi.qualys.com/api/2.0/fo/scan/compliance/complianc
e_scan_result_output.dtd">
<COMPLIANCE_SCAN_RESULT_OUTPUT>
  <RESPONSE>
    <DATETIME>2012-09-17T10:23:53Z</DATETIME>
```

```
    <COMPLIANCE_SCAN>
      <HEADER>
        <NAME><![CDATA[Compliance Scan Results]]></NAME>
        <GENERATION_DATETIME>2012-09-
17T10:23:53Z</GENERATION_DATETIME>
        <COMPANY_INFO>
          <NAME><![CDATA[Qualys]]></NAME>
          <ADDRESS><![CDATA[1600 Bridge Parkway]]></ADDRESS>
          <CITY><![CDATA[Redwood Shores]]></CITY>
          <STATE><![CDATA[California]]></STATE>
          <COUNTRY><![CDATA[United States]]></COUNTRY>
          <ZIP_CODE><![CDATA[94065]]></ZIP_CODE>
        </COMPANY_INFO>
        <USER_INFO>
          <NAME><![CDATA[NAME]]></NAME>
          <USERNAME>USERNAME</USERNAME>
          <ROLE>Manager</ROLE>
        </USER_INFO>
        <KEY value="USERNAME">USERNAME</KEY>
        <KEY value="COMPANY"><![CDATA[Qualys]]></KEY>
        <KEY value="DATE">2012-09-15T11:49:08Z</KEY>
        <KEY value="TITLE"><![CDATA[My PC Scan]]></KEY>
        <KEY value="TARGET">10.10.10.29</KEY>
        <KEY value="EXCLUDED_TARGET"><![CDATA[N/A]]></KEY>
        <KEY value="DURATION">00:01:00</KEY>
        <KEY value="SCAN_HOST">10.10.21.122 (Scanner 6.6.28-1,
Vulnerability Signatures 2.2.215-

2)</KEY>
        <KEY value="NBHOST_ALIVE">1</KEY>
        <KEY value="NBHOST_TOTAL">1</KEY>
        <KEY value="REPORT_TYPE">Scheduled</KEY>
        <KEY value="OPTIONS">File Integrity Monitoring: Enabled,
Scanned Ports: Standard Scan, Hosts to Scan in Parallel - External
Scanners: 15, Hosts to Scan in Parallel - Scanner Appliances: 30,
Total Processes to Run in Parallel: 10, HTTP Processes to Run in
Parallel: 10,

Packet (Burst) Delay: Medium, Intensity: Normal, Overall
Performance: Normal, ICMP Host Discovery, Ignore RST packets: Off,
Ignore firewall-generated SYN-ACK packets: Off, Do not send ACK or
SYN-ACK packets during host discovery: Off</KEY>
```

```
        <KEY value="STATUS">FINISHED</KEY>
        <OPTION_PROFILE>
          <OPTION_PROFILE_TITLE
option_profile_default="0"><![CDATA[11412]]
></OPTION_PROFILE_TITLE>
        </OPTION_PROFILE>
      </HEADER>
      <APPENDIX>
        <TARGET_HOSTS>
          <HOSTS_SCANNED>10.10.10.29</HOSTS_SCANNED>
        </TARGET_HOSTS>
        <TARGET_DISTRIBUTION>
          <SCANNER>
            <NAME><![CDATA[iscan_sx]]></NAME>
            <HOSTS>10.10.10.29</HOSTS>
          </SCANNER>
        </TARGET_DISTRIBUTION>
        <AUTHENTICATION>
          <AUTH>
            <TYPE>Windows</TYPE>
            <SUCCESS>
              <IP>10.10.10.29</IP>
            </SUCCESS>
          </AUTH>
        </AUTHENTICATION>
      </APPENDIX>
    </COMPLIANCE_SCAN>
  </RESPONSE>
</COMPLIANCE_SCAN_RESULT_OUTPUT>
```

Scan Schedules

The Scan Schedule API v2 (**`/api/2.0/fo/schedule/scan/`**) supports defining schedules for vulnerability scans. This API delivers improvements to the API v1 (/msp/scheduled_scans.php) and supports scanning targets in multiple network zones.

Permissions

User Role	Permissions
Manager	Create scan schedules for all assets in the subscription Remove all scan schedules View all scan schedules in the subscription
Unit Manager	Create scan schedules for assets in user's business unit Remove scan schedules in user's business unit. View scan schedules in the subscription*
Scanner	Create scan schedules for assets in user's account. Remove user's scan schedules View scan schedules in the subscription*
Readers	No permission to create or remove scan schedules View scan schedules in the subscription*

* Qualys includes an account permission setting that restricts Unit Managers, Scanners, and Readers from viewing scheduled tasks on unassigned assets.

List scan schedules

Use these parameters:

Parameter	Description
action=list	(Required) Supported method: GET
echo_request={**0**|1}	(Optional) Specify 1 to echo the request's input parameters (names and values) in the XML output. Otherwise parameters are not displayed in the output.
id={value}	(Optional) The ID of the scan schedule you want to display.
active={0|1}	(Optional) Specify 1 for active schedules only, or 0 for deactivated schedules only.

API request:

```
curl u "USERNAME:PASSWORD" -H "X-Requested-With: Curl"
"https://qualysapi.qualys.com/api/2.0/fo/schedule/scan/?action=lis
t&id=85331&echo_request=1"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SCHEDULE_SCAN_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/schedule/scan/schedule_sc
an_list_output.dtd">
<SCHEDULE_SCAN_LIST_OUTPUT>
  <REQUEST>
    <DATETIME>2015-02-13T08:33:04Z</DATETIME>
    <USER_LOGIN>acme_ac</USER_LOGIN>
<RESOURCE>https://qualysapi.qualys.com/api/2.0/fo/schedule/scan/</
RESOURCE>
    <PARAM_LIST>
      <PARAM>
        <KEY>action</KEY>
        <VALUE>list</VALUE>
      </PARAM>
      <PARAM>
        <KEY>id</KEY>
        <VALUE>1128216540</VALUE>
      </PARAM>
      <PARAM>
        <KEY>echo_request</KEY>
        <VALUE>1</VALUE>
      </PARAM>
    </PARAM_LIST>
  </REQUEST>
  <RESPONSE>
    <DATETIME>2015-02-13T08:33:04Z</DATETIME>
    <SCHEDULE_SCAN_LIST>
      <SCAN>
        <ID>1128216540</ID>
        <ACTIVE>1</ACTIVE>
        <TITLE><![CDATA[Schedule - (GMT -05:00) Brazil, Amazonas
(far west)]]></TITLE>
        <USER_LOGIN>acme_ac</USER_LOGIN>
        <TARGET><![CDATA[10.10.10.2-10.10.10.30]]></TARGET>
```

```
        <NETWORK_ID><![CDATA[0]]></NETWORK_ID>
        <ISCANNER_NAME><![CDATA[External
Scanner]]></ISCANNER_NAME>
        <USER_ENTERED_IPS>
          <RANGE>
            <START>10.10.10.2</START>
            <END>10.10.10.30</END>
          </RANGE>
        </USER_ENTERED_IPS>
        <OPTION_PROFILE>
          <TITLE><![CDATA[Authenticated Scan v.1]]></TITLE>
          <DEFAULT_FLAG>0</DEFAULT_FLAG>
        </OPTION_PROFILE>
        <SCHEDULE>
          <MONTHLY frequency_months="1"
day_of_month="12"></MONTHLY>
          <START_DATE_UTC>2015-02-12T18:10:00Z</START_DATE_UTC>
          <START_HOUR>13</START_HOUR>
          <START_MINUTE>10</START_MINUTE>
          <PAUSE_AFTER_HOURS></PAUSE_AFTER_HOURS>
          <RESUME_IN_DAYS></RESUME_IN_DAYS>
          <NEXTLAUNCH_UTC>2015-03-12T17:10:00</NEXTLAUNCH_UTC>
          <TIME_ZONE>
            <TIME_ZONE_CODE>BR-AM1</TIME_ZONE_CODE>
            <TIME_ZONE_DETAILS>(GMT-0500) Brazil:
America/Manaus</TIME_ZONE_DETAILS>
          </TIME_ZONE>
          <DST_SELECTED>0</DST_SELECTED>
          <MAX_OCCURRENCE>3</MAX_OCCURRENCE>
        </SCHEDULE>
      </SCAN>
    </SCHEDULE_SCAN_LIST>
  </RESPONSE>
</SCHEDULE_SCAN_LIST_OUTPUT>
```

DTD:

The scheduled scan list output DTD can be found at the following URL (when your account is located on US Platform 1):

```
https://qualysapi.qualys.com/api/2.0/fo/schedule/scan/schedule_sca
n_list_output.dtd
```

Create a scan schedule

The input parameters for creating a scan schedule are below. For complete details see "Scan Parameters" and "Scan Schedule Parameters".

Type	Parameter List
Request	action=create (required), echo_request
Scan	scan_title (required), active=0|1 (required)
Option Profile	option_id or option_profile (one is required)
Scanner Appliance	iscanner_id or iscanner_name
Asset IPs/Groups	ip, asset_group_ids, asset_groups, exclude_ip_per_scan, default_scanner, scanners_in_ag
Asset Tags	target_from=tags, tag_include_selector, tag_exclude_selector, tag_set_by, tag_set_exclude, tag_set_include, use_ip_nt_range_tags
Network	ip_network_id to filter IPs/ranges in "ip" parameter (valid when the networks feature is enabled)
EC2 Hosts	target_from=tags (required) use_ip_nt_range_tags=0 (optional) tag_set_include (required) More Asset Tags parameters (optional)
EC2 Environment	connector_name or connector_uuid (one is required) ec2_endpoint (required)
Scheduling	start_date (current date by default) start_hour, start_minute, time_zone_code, occurrence (required) observe_dst, recurrence, end_after, pause_after_hours, resume_in_days
Daily Scan	occurrence=daily, frequency_days (required)
Weekly Scan	occurrence=weekly, frequency_weeks, weeks (required)
Monthly Scan	occurrence=monthly, frequency_months (required) Nth day of month: day_of_month (required) Day in Nth week: day_of_week, week_of_month (required)

API request:

```
curl -u "USERNAME:PASSWD" -H "X-Requested-With: curl" -X "POST" -d
"scan_title=My+Scan+Schedule&active=1&option_id=3456&target_from=t
ags&tag_set_include=tag1,tag2,tag3&iscanner_name=scanner1&occurren
ce=daily&frequency_days=5&time_zone_code=US-CA&observe_dst=yes&sta
rt_hour=14&start_minute=0"
"https://qualysapi.qualys.com/api/2.0/fo/schedule/scan/?action=cre
ate"
```

XML output:

```
?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2015-01-20T21:32:40Z</DATETIME>
    <TEXT>New scan scheduled successfully</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>136992</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

Delete a scan schedule

Use these parameters:

Parameter	Description
action=delete	(Required) Supported method: POST
echo_request={**0**|1}	(Optional) Specify 1 to echo the request's input parameters (names and values) in the XML output. Otherwise parameters are not displayed in the output.
id={value}	(Optional) The ID of the scan schedule you want to delete.

API request:

```
curl -u "USERNAME:PASSWD" -H "X-Requested-With: curl" -X "POST" -d
"id=123456"
"https://qualysapi.qualys.com/api/2.0/fo/schedule/scan/?action=del
ete"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2015-01-30T21:32:40Z</DATETIME>
    <TEXT>Schedule scan deleted successfully</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>123456</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

Scan List Parameters

Request Type

The scan list type parameters are described below.

Parameter	Description
action=list	(Required) A flag used to make a request for a scan list.
echo_request={0|1}	(Optional) Specifies whether to echo the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.

Scan List Filters

Several parameters allow you to set filters to restrict the scan list output. When no filters are specified, the service returns all scans launched by all users within the past 30 days.

Parameter	Description
scan_ref={value}	(Optional) Show only a scan with a certain scan reference code. When unspecified, the scan list is not restricted to a certain scan. For a vulnerability scan, the format is: scan/987659876.19876 For a compliance scan the format is: compliance/98765456.12345 For a SCAP scan the format is: qscap/987659999.22222
scan_id={value}	(Optional) Show only a scan with a certain compliance scan ID.
state={value}	(Optional) Show only one or more scan states. By default, the scan list is not restricted to certain states. A valid value is: Running, Paused, Canceled, Finished, Error, Queued (scan job is waiting to be distributed to scanner(s)), or Loading (scanner(s) are finished and scan results are being loaded onto the platform). Multiple values are comma separated.
processed={0|1}	(Optional) Specify 0 to show only scans that are not processed. Specify 1 to show only scans that have been processed. When not specified, the scan list output is not filtered based on the processed status.
type={value}	(Optional) Show only a certain scan type. By default, the scan list is not restricted to a certain scan type. A valid value is: On-Demand, Scheduled, or API.

Parameter	Description
target={value}	(Optional) Show only one or more target IP addresses. By default, the scan list includes all scans on all IP addresses. Multiple IP addresses and/or ranges may be entered. Multiple entries are comma separated. You may enter an IP address range using the hyphen (-) to separate the start and end IP address, as in: 10.10.10.1-10.10.10.2
user_login={value}	(Optional) Show only a certain user login. The user login identifies a user who launched scans. By default, the scan list is not restricted to scans launched by a particular user. Enter the login name for a valid Qualys user account.
launched_after_datetime={date}	(Optional) Show only scans launched after a certain date and time (optional). The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT), like "2007-07-01" or "2007-01-25T23:12:00Z". When **launched_after_datetime** and **launched_before_datetime** are unspecified, the service selects scans launched within the past 30 days. A date/time in the future returns an empty scans list.
launched_before_datetime={date}	(Optional) Show only scans launched before a certain date and time (optional). The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT), like "2007-07-01" or "2007-01-25T23:12:00Z". When **launched_after_datetime** and **launched_before_datetime** are unspecified, the service selects scans launched within the past 30 days. A date/time in the future returns a list of all scans (not limited to scans launched within the past 30 days).

Show in Scan List

These parameters specify whether certain information will be shown in the XML output.

Parameter	Description
show_ags={**0**|1}	(Optional) Specify 1 to show asset group information for each scan in the XML output. By default, asset group information is not shown.
show_op={**0**|1}	(Optional) Specify 1 to show option profile information for each scan in the XML output. By default, option profile information is not shown.

Parameter	Description
show_status={0 | **1**}	(Optional) Specify 0 to not show scan status for each scan in the XML output. By default, scan status is shown.
show_last={**0** | 1}	(Optional) Specify 1 to show only the most recent scan (which meets all other search filters in the request) in the XML output. By default, all scans are shown in the XML output.
pci_only={**0** | 1}	(Optional) Specify 1 to show only external PCI scans in the XML output. External PCI scans are vulnerability scans run with the option profile "Payment Card Industry (PCI) Options". When **pci_only=1** is specified, the XML output will not include other types of scans run with other option profiles.

Scan Parameters

Input parameters used to launch a VM or PC scan are described below.

Parameter	Description
action={launch}	(Required) Specify "launch" to launch a new scan.
echo_request={**0** | 1}	(Optional) Specify 1 to list the input parameters in the XML output. When unspecified, parameters are not listed in the XML output.
scan_title={value}	(Optional) The scan title. This can be a maximum of 2000 characters (ascii).
target_from={**assets** | tags}	(Optional) Specify "assets" (the default) when your scan target will include IP addresses/ranges and/or asset groups. Specify "tags" when your scan target will include asset tags.
ip={value}	(Optional) The IP addresses to be scanned. You may enter individual IP addresses and/or ranges. Multiple entries are comma separated. One of these parameters is required: ip, asset_groups or asset_group_ids.
	ip is valid only when target_from=assets is specified.
asset_groups={value}	(Optional) The titles of asset groups containing the hosts to be scanned. Multiple titles are comma separated. One of these parameters is required: ip, asset_groups or asset_group_ids.
	asset_groups is valid only when target_from=assets is specified. These parameters are mutually exclusive and cannot be specified in the same request: asset_groups and asset_group_ids.
asset_group_ids={value}	(Optional) The IDs of asset groups containing the hosts to be scanned. Multiple IDs are comma separated. One of these parameters is required: ip, asset_groups or asset_group_ids.
	asset_group_ids is valid only when target_from=assets is specified. These parameters are mutually exclusive and cannot be specified in the same request: asset_groups and asset_group_ids.
exclude_ip_per_scan={value}	(Optional) The IP addresses to be excluded from the scan when the scan target is specified as IP addresses (not asset tags). You may enter individual IP addresses and/or ranges. Multiple entries are comma separated.
	exclude_ip_per_scan is valid only when target_from=assets is specified.

Parameter	Description
tag_include_selector={all | **any**}	(Optional) Select "any" (the default) to include hosts that match at least one of the selected tags. Select "all" to include hosts that match all of the selected tags.
	tag_include_selector is valid only when target_from=tags is specified.
tag_exclude_selector={all | **any**}	(Optional) Select "any" (the default) to exclude hosts that match at least one of the selected tags. Select "all" to exclude hosts that match all of the selected tags.
	tag_exclude_selector is valid only when target_from=tags is specified.
tag_set_by={**id** | name}	(Optional) Specify "id" (the default) to select a tag set by providing tag IDs. Specify "name" to select a tag set by providing tag names.
	tag_set_by is valid only when target_from=tags is specified.
tag_set_include={value}	(Optional) Specify a tag set to include. Hosts that match these tags will be included. You identify the tag set by providing tag name or IDs. Multiple entries are comma separated.
	tag_set_include is valid only when target_from=tags is specified.
tag_set_exclude={value}	(Optional) Specify a tag set to exclude. Hosts that match these tags will be excluded. You identify the tag set by providing tag name or IDs. Multiple entries are comma separated.
	tag_set_exclude is valid only when target_from=tags is specified.
use_ip_nt_range_tags={**0** | **1**}	(Optional) Specify "0" (the default) to select from all tags (tags with any tag rule). Specify "1" to scan all IP addresses defined in tags. When this is specified, only tags with the dynamic IP address rule called "IP address in Network Range(s)" can be selected.
	use_ip_nt_range_tags is valid only when target_from=tags is specified.
iscanner_id={value}	(Optional) The IDs of the scanner appliances to be used. Multiple entries are comma separated. For an Express Lite user, Internal Scanning must be enabled in the user's account.
	One of these parameters must be specified in a request: iscanner_name, iscanner_id, default_scanner, scanners_in_ag. These parameters are mutually exclusive and cannot be specified in the same request: iscanner_id and iscanner_name.

Parameter	Description
iscanner_name={value}	(Optional) The friendly names of the scanner appliances to be used. Multiple entries are comma separated. For an Express Lite user, Internal Scanning must be enabled in the user's account.
	One of these parameters must be specified in a request for an internal scan: iscanner_name, iscanner_id, default_scanner, scanners_in_ag. When none of these are specifed, the External scanners are used. These parameters are mutually exclusive and cannot be specified in the same request: iscanner_id and iscanner_name.
default_scanner={**0**|1}	(Optional) Specify 1 to use the default scanner in each target asset group. For an Express Lite user, Internal Scanning must be enabled in the user's account.
	One of these parameters must be specified in a request for an internal scan: iscanner_name, iscanner_id, default_scanner, scanners_in_ag. When none of these are specifed, the External scanners are used. default_scanner is valid when the scan target is specified using one of these parameters: asset_groups, asset_group_ids.
scanners_in_ag={**0**|1}	(Optional) Specify 1 to distribute the scan to the target asset groups' scanner appliances (pool of 5 appliances per group). Appliances in each asset group are tasked with scanning the IPs in the group. For an Express Lite user, Internal Scanning must be enabled in the user's account.
	One of these parameters must be specified in a request for an internal scan: iscanner_name, iscanner_id, default_scanner, scanners_in_ag. When none of these are specifed, the External scanners are used. scanners_in_ag is valid when the scan target is specified using one of these parameters: asset_groups, asset_group_ids.
option_title={value}	(Optional) The title of the compliance option profile to be used.
	One of these parameters must be specified in a request: option_title or option_id. These are mutually exclusive and cannot be specified in the same request.
option_id={value}	(Optional) The ID of the compliance option profile to be used.
	One of these parameters must be specified in a request: option_title or option_id. These are mutually exclusive and cannot be specified in the same request.
connector_name={value}	(Required for an EC2 scan) The name of the EC2 connector for the AWS integration you want to run the scan on.

Parameter	Description
ec2_endpoint={value}	(Required for an EC2 scan) The EC2 region code or the ID of the Virtual Private Cloud (VPC) zone. Need help finding the region code? See the following: http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html#concepts-regions-availability-zones
ip_network_id={value}	(Optional, and valid only when the Network Support feature is enabled for the user's account) The ID of a network used to filter the IPs/ranges specified in the"ip" parameter. Set to a custom network ID (note this does not filter IPs/ranges specified in "asset_groups" or "asset_group_ids"). Or set to "0" (the default) for the Global Default Network - this is used to scan hosts outside of your custom networks.
runtime_http_header={value}	Set a custom value in order to drop defenses (such as logging, IPs, etc) when an authorized scan is being run. The value you enter will be used in the "Qualys-Scan:" header that will be set for many CGI and web application fingerprinting checks. Some discovery and web server fingerprinting checks will not use this header.

Scan Schedule Parameters

Scan Schedule - Occurrence

Parameter	Description
`occurrence=daily`	Required for a daily scan.
frequency_days={value}	Required for a daily scan. The scan will run every N number of days. Value is an integer from 1 to 365.
`occurrence=weekly`	Required for a weekly scan.
frequency_weeks={value}	Required for a weekly scan. The scan will run every N number of weeks. Value is an integer from 1 to 52.
weekdays={value}	Required for a weekly scan. The scan will run on the one or more weekdays. Value is one or more days: sunday, monday, tuesday, wednesday, thursday, friday, saturday. Multiple days are comma separated.
`occurrence=monthly`	Required for a monthly scan.
frequency_months={value}	Required for a monthly scan. The scan will run every N number of months. Value is an integer from 1 to 12.
day_of_month={value}	Required for monthly scan - Nth day of the month. The scan will run on the Nth day of the month. Value is an integer from 1 to 31.
day_of_week={value}	Required for monthly scan - day in Nth week. The scan will run on this day of the week. Value is and integer from 0 to 6, where 0 is Sunday and 2 is Tuesday.
week_of_month={value}	Required for monthly scan - day in Nth week. The scan will run on this week of the month. Value is one of: first, second, third, fourth, last.

Scan Schedule - Start Time

Parameter	Description
start_date={mm/dd/yyyy}	(Optional) By default the start date is the date when the schedule is created. You can define another start date in mm/dd/yyyy format.
start_hour={hour}	(Required) The hour when a scan will start. The hour is an integer from 0 to 23, where 0 represents 12 AM, 7 represents 7 AM, and 22 represents 10 PM.
start_minute={minute}	(Required) The minute when a scan will start. A valid value is an integer from 0 to 59.

Parameter	Description
time_zone_code={value}	(Required) The time zone code for starting a scan, in upper case. For example, the time zone code for US California is US-CA. Valid codes are returned by the Time Zone Code API (/msp/time_zone_code_list.php).
observe_dst={yes | **no**}	(Optional) Specify yes to observe Daylight Saving Time (DST). This parameter is valid when the time zone code specified in time_zone_code supports DST.
recurrence={value}	(Optional) The number of times the scan will be run before it is deactivated. For example, if you set recurrence=2, the scan schedule will be deactivated after it runs 2 times. By default no value is set. A valid value is an integer from 1 to 99.
end_after_hours={value}	(Optional) End a scan after some number of hours. A valid value is an integer from 1 to 48. By default our service does not end a scan after some number of hours.
pause_after_hours={value}	(Optional) Pause a scan after some number of hours if the scan has not finished by then. A valid value is an integer from 1 to 120.
resume_in_days={value}	(Optional) Resume a paused scan in some number of days. A valid value is an integer from 1 to 9.

Scanner Appliances

The Scanner Appliance API v2 (**/api/2.0/fo/appliance/**) is used to get a list of the scanner appliances in your account - both physical and virtual - and to manage VLANs and static routes for appliances in your account. See Chapter 2, "Authentication Using the V2 APIs."

Express Lite: This API is available to Express Lite users when Internal Scanning is enabled in the user's account.

Permissions for using the Scanner Appliance API v2:

User Role	Permissions
Manager	View all scanner appliances in the subscription. Manage (add, remove) VLANs and static routes for all appliances.
Unit Manager	View scanner appliances in user's business unit. Manage (add, remove) VLANs and static routes for appliances in their assigned business unit.
Scanner	View scanner appliances in user's account.
Reader	View scanner appliances in user's account.

Scanner Appliance List

The Scanner Appliance List API v2 (**/api/2.0/fo/appliance/** with **action=list**) is used to list scanner appliances in your account with their configurations. The list output is shown in "brief" mode by default. Specify **output_mode=full** to include full output (the same information available within the Qualys user interface).

The parameters used to request a scanner appliance list are described below.

Parameter	Description
action=list	(Required) A flag used to make a request for a list of scanner appliances. The GET or POST method may be used for a list request.
echo_request={**0** | 1}	(Optional) Specifies whether to echo the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.

Parameter	Description
output_mode={**brief** | full}	(Optional) The amount of detail provided for each scanner appliance in the output: brief (default) or full. The "brief" output includes this information for each appliance: appliance ID, friendly name, software version, the number of running scans, and heartbeat check status (online or offline). The "full" output includes the full appliance information, including the same details available in the Qualys user interface.
scan_detail={**0** | 1}	(Optional) Specifies whether to include scan details for scans currently running on the scanner appliance. Specify 1 to include scan details. Specify 0 to not include scan details. Scan detail includes: scan ID, title, scan reference, scan type and scan date.
busy={0 | 1}	(Optional) By default all scanner appliances in the user account are shown. Specify 0 to show only appliances which are not currently running scans. Specify 1 to show only appliances which are currently running scans.
scan_ref={value}	(Optional) Specify a scan reference code to show only the scanner appliances running a particular scan.You may enter a valid scan reference code for a currently running scan. The scan reference code starts with a string that identifies the scan type: "scan/" for a vulnerability scan, "compliance/" for a compliance scan, "was/" for a web application scan, "qscap/" for an FDCC scan, or "map/" for a network map.
name={string}	(Optional) List only scanner appliances (physical and virtual) that have names matching the string provided. Tip - Substring match is supported. For example, if you have 2 appliances named "myscanner" and "anotherscanner" and you supply the string "name=scan" both appliance both appliances will be returned in the XML output.
ids={id1,id2,..}	(Optional) List only scanner appliances (physical and virtual) that have certain IDs. Multiple IDs are comma separated.
include_license_info={**0** | 1}	(Optional) Set to 1 to return virtual scanner license information in the XML output. This tells you the number of licenses you have and the number used. This information is not returned by default. When specified the XML output will include the LICENSE_INFO element.

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -d
"action=list&echo_request=1&ids=777,1127,1131&include_license_info
=1" "https://qualysapi.qualys.com/api/2.0/fo/appliance/"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE APPLIANCE_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/appliance/appliance_list_
output.dtd">
<APPLIANCE_LIST_OUTPUT>
    <RESPONSE>
        <DATETIME>2014-01-02T09:26:01Z</DATETIME>
        <APPLIANCE_LIST>
            <APPLIANCE>
                <ID>777</ID>
                <NAME>scanner1</NAME>
                <SOFTWARE_VERSION>2.6</SOFTWARE_VERSION>
                <RUNNING_SCAN_COUNT>0</RUNNING_SCAN_COUNT>
                <STATUS>Online</STATUS>
            </APPLIANCE>
            <APPLIANCE>
                <ID>1127</ID>
                <NAME>scanner2</NAME>
                <SOFTWARE_VERSION>2.6</SOFTWARE_VERSION>
                <RUNNING_SCAN_COUNT>0</RUNNING_SCAN_COUNT>
                <STATUS>Online</STATUS>
            </APPLIANCE>
            <APPLIANCE>
                <ID>1131</ID>
                <NAME>scanner3</NAME>
                <SOFTWARE_VERSION>2.6</SOFTWARE_VERSION>
                <RUNNING_SCAN_COUNT>0</RUNNING_SCAN_COUNT>
                <STATUS>Offline</STATUS>
            </APPLIANCE>
        </APPLIANCE_LIST>
        <LICENSE_INFO>
            <QVSA_LICENSES_COUNT>10</QVSA_LICENSES_COUNT>
            <QVSA_LICENSES_USED>3</QVSA_LICENSES_USED>
        </LICENSE_INFO>
    </RESPONSE>
```

```
</APPLIANCE_LIST_OUTPUT>
```

DTD:

An API request to view a list of scanner appliances returns XML output using the "appliance_list_output.dtd" which can be found at the following URL (where <qualysapi.qualys.com> identifies the Qualys API server URL where your account is located):

```
https://<qualysapi.qualys.com>/api/2.0/fo/appliance/
appliance_list_output.dtd
```

The DTD for "appliance_list_output.dtd" is provided in Appendix A.

Manage VLANs and Static Routes

Manage your VLANs and static routes for virtual scanner appliances using the Virtual Scanner Appliance API (api/2.0/fo/appliance/?action=update). Use the new parameters "set_vlans" and "set_routes" to add, update and remove these setttings.

What do I need? Your Qualys account must have the VLANs and Static Routes feature enabled. Please contact our Support Team or your Qualys TAM if you would like us to enable this feature for you.

Set VLANs on Scanner Appliance

Use the "set_vlans" parameter to specify one or more VLANs.

The format for a single VLAN is ID|IP_ADDRESS|NETMASK|NAME, with pipe (|) used as a delimiter. All attributes are required (ID, IP_ADDRESS, etc). Multiple VLANs can be assigned using a comma separated list.

Good to know An API call with the parameter "set_vlans" set to " (empty string) will replace (i.e. remove) *all* of the VLANs that are assigned to the scanner appliance.

Attribute	Description
ID	Customer-defined ID (not assigned by Qualys). Must be in the range 0 to 4096, inclusive.
IP_ADDRESS	A valid IPv4 IP address (dotted quad), such as 10.10.10.1
NETMASK	A valid network mask (dotted quad), such as 255.255.255.0
NAME	A valid name (can be empty). The name can be a maximum of 256 ASCII characters. The charater : (colon) is permitted. These characters are not permitted: , (comma), < (less than), > (greater than), " (double quote), & (ampersand), |(pipe), = (equals).

API request (1 VLAN):

```
curl -u "USERNAME:PASSWD" -H "X-Requested-With: Curl" -X "POST" -d
"id=43463&set_vlans=0|10.10.10.1|255.255.255.0|vlan1"
"https://qualysapi.qualys.com/api/2.0/fo/appliance/?action=update"
```

API request (multiple VLANs):

```
curl -u "USERNAME:PASSWD" -H "X-Requested-With: Curl" -X "POST" -d
"id=43463&set_vlans=1|10.10.10.1|255.255.255.0|vlan1,2|10.10.10.2|
255.255.255.0|vlan2"
"https://qualysapi.qualys.com/api/2.0/fo/appliance/?action=update"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2014-07-09T08:46:54Z</DATETIME>
    <TEXT>Virtual scanner updated successfully</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>43463</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

Set Static Routes on Scanner Appliance

Use the “set_routes” parameter to specify one or more static routes. The format for a single static route is IP_ADDRESS|NETMASK|GATEWAY|NAME, with pipe (|) used as the delimiter. All attributes are required (IP_ADDRESS, NETMASK, etc). Multiple static routes can be assigned using a comma separated list.

Good to know An API call with the parameter "set_routes" set to " (empty string) will replace (i.e. remove) *all* of the static routes that are assigned to the scanner appliance.

Attribute	Description
IP_ADDRESS	A valid IPv4 IP address (dotted quad), such as 10.10.26.0
NETMASK	A valid network mask (dotted quad), such as 255.255.255.0
GATEWAY	A valid IPv4 address (dotted quad), such as 10.10.25.255
NAME	A valid name (can be empty). The name can be a maximum of 256 ASCII characters. The charater : (colon) is permitted. These characters are not permitted: , (comma), < (less than), > (greater than), " (double quote), & (ampersand), |(pipe), = (equals).

API request (1 static route):

```
curl -u "USERNAME:PASSWD" -H "X-Requested-With: Curl" -X "POST" -d
"id=43463&set_routes=10.10.25.0|255.255.255.0|10.10.25.255|SRoute1
"
"https://qualysapi.qualys.com/api/2.0/fo/appliance/?action=update"
```

API request (multiple static routes):

```
curl -u "USERNAME:PASSWD" -H "X-Requested-With: Curl" -X "POST" -d
"id=43463&set_routes=10.10.25.0|255.255.255.0|10.10.25.255|Route1,
10.10.26.0|255.255.255.0|10.10.26.255|Route2"
"https://qualysapi.qualys.com/api/2.0/fo/appliance/?action=update"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2014-07-09T08:49:18Z</DATETIME>
    <TEXT>Virtual scanner updated successfully</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>43463</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
```

```
</SIMPLE_RETURN>
```

View Scanner Appliances with VLANs, Static Routes

Use the parameters "action=list" and "output_mode=full".

API request:

```
curl -u "USERNAME:PASSWD" -H "X-Requested-With:
"https://qualysapi.qualys.com/api/2.0/fo/appliance/?action=list&id
s=43463&output_mode=full"
```

XML output:

```
...
   <VLANS>
      <SETTING>Enabled</SETTING>
      <VLAN>
         <ID>0</ID>
         <NAME>vlan1</NAME>
         <IP_ADDRESS>10.10.10.1</IP_ADDRESS>
         <NETMASK>255.255.255.0</NETMASK>
      </VLAN>
   </VLANS>
   <STATIC_ROUTES>
      <ROUTE>
         <NAME>Route1</NAME>
         <IP_ADDRESS>10.10.25.0</IP_ADDRESS>
         <NETMASK>255.255.255.0</NETMASK>
         <GATEWAY>10.10.25.255</GATEWAY>
      </ROUTE>
      <ROUTE>
         <NAME>Route2</NAME>
         <IP_ADDRESS>10.10.26.0</IP_ADDRESS>
         <NETMASK>255.255.255.0</NETMASK>
         <GATEWAY>10.10.26.255</GATEWAY>
      </ROUTE>
   </STATIC_ROUTES>
...
```

Delete All VLAN Records

Use the "set_vlans" parameters and set it to " (empty string).

API request (deletes all VLAN records):

```
curl -u "USERNAME:PASSWD" -H "X-Requested-With: -d
"id=43463&set_vlans="
"https://qualysapi.qualys.com/api/2.0/fo/appliance/?action=update"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2014-07-09T08:49:18Z</DATETIME>
    <TEXT>Virtual scanner updated successfully</TEXT>
...
```

Delete All Static Route Records

Use the "set_routes" parameters and set it to " (empty string).

API request (deletes all static route records):

```
curl -u "USERNAME:PASSWD" -H "X-Requested-With: -d
"id=43463&set_routes="
"https://qualysapi.qualys.com/api/2.0/fo/appliance/?action=update"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2014-07-09T08:49:18Z</DATETIME>
    <TEXT>Virtual scanner updated successfully</TEXT>
...
```

Virtual Scanner Appliances

Using the Scanner Appliance API v2 (**`/api/2.0/fo/appliance/`**) Managers and Unit Managers can create, update and delete virtual scanner appliances.

Tell me about permissions. Managers can perform all actions (create, update, delete). Unit Managers can perform these actions on virtual scanners in their business unit. Scanners and Readers do not have permissions to create, update and delete virtual scanners.

Add New Virtual Scanner

Use these parameters:

Parameter	Description
action=create	(Required) The POST method must be used.
name={string}	(Required) The friendly name. This name can't already be assigned to an appliance in your account. It can be a maximum of 15 characters, spaces are not allowed.
polling_interval={value}	(Optional) The polling interval, in seconds. A valid value is 60 to 3600 (we recommend 180 which is the default). This is the frequency that the virtual scanner will attempt to connect to our Cloud Security Platform. The appliance calls home to provide health updates/heartbeats to the platform, to get software updates from the platform, to learn if new scan jobs have been requested by users, and to upload scan results data to the platform, if applicable.

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST"
-d "action=create&echo_request=1&name=scanner1"
"https://qualysapi.qualys.com/api/2.0/fo/appliance/"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE APPLIANCE_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/appliance/appliance_creat
e_output.dtd">
<APPLIANCE_CREATE_OUTPUT>
    <RESPONSE>
        <DATETIME>2014-01-02T09:26:01Z</DATETIME>
        <ID>777</ID>
```

```
        <NAME>scanner1</NAME>
        <ACTIVATION CODE>ACTIVATION-CODE</ACTIVATION CODE>
        <REMAINING_QVSA_LICENSES>4</REMAINING_QVSA_LICENSES>
    </RESPONSE>
</APPLIANCE_CREATE_OUTPUT>
```

DTD for XML output:

The DTD for the XML output can be found here (where qualysapi.qualys.com is the base URL for your Qualys platform:

```
https://qualysapi.qualys.com/api/2.0/fo/appliance/appliance_create
_output.dtd
```

Update a Virtual Scanner

Use these parameters:

Parameter	Description
action=update	(Required) The POST method must be used.
id={id}	(Required) A valid ID of a virtual scanner.
name={string}	(Optional) The friendly name. This name can't already be assigned to an appliance in your account. It can be a maximum of 15 characters, spaces are not allowed.
polling_interval={value}	(Optional) The polling interval, in seconds. A valid value is 60 to 3600 (we recommend 180 which is the default). This is the frequency that the virtual scanner will attempt to connect to our Cloud Security Platform. The appliance calls home to provide health updates/heartbeats to the platform, to get software updates from the platform, to learn if new scan jobs have been requested by users, and to upload scan results data to the platform, if applicable.
comment={value}	(Optional) User-defined comments.

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST"
-d "action=update&echo_request=1&id=12345&name=scanner15"
"https://qualysapi.qualys.com/api/2.0/fo/appliance/"
```

XML output:

The XML output uses the simple return (/api/2.0/simple_return.dtd).

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
    <RESPONSE>
        <DATETIME>2014-04-03T12:12:45Z</DATETIME>
        <TEXT>Virtual scanner updated successfully</TEXT>
        <ITEM_LIST>
            <ITEM>
                <KEY>ID</KEY>
                <VALUE>17110</VALUE>
            </ITEM>
        </ITEM_LIST>
    </RESPONSE>
</SIMPLE_RETURN>
```

Delete a Virtual Scanner

Deleting a virtual scanner appliance results in these actions: 1) The virtual scanner will be removed from associated Asset Groups, and 2) Scheduled Scans using this virtual scanner will be deactivated.

Is your virtual scanner running scans? If yes it's not possible to delete it. We recommend you check to be sure the virtual scanner you want to delete is not running scans.

Use these parameters:

Parameter	Description
action=delete	(Required) The POST method must be used.
id={id}	(Required) A valid ID of a virtual scanner.

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST"
-d "action=delete&echo_request=1&id=12345"
"https://qualysapi.qualys.com/api/2.0/fo/appliance/"
```

XML output:

The XML output uses the simple return (/api/2.0/simple_return.dtd). If schedules and/or asset groups were impacted we'll list them so you can update them as needed.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE APPLIANCE_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/simple_return.dtd">
<SIMPLE_RETURN>
    <RESPONSE>
        <DATETIME>2014-01-02T09:26:01Z</DATETIME>
        <TEXT>Virtual scanner deleted successfully</ID>
        <ITEM_LIST>
             <ITEM>
                  <KEY>ID<KEY>
                  <VALUE>115<VALUE>
             </ITEM>
             <ITEM>
                  <KEY>DEACTIVATED_SCHEDULED_SCANS<KEY>
                  <VALUE>None<VALUE>
             </ITEM>
             <ITEM>
                  <KEY>AFFECTED_ASSET_GROUPS<KEY>
                  <VALUE>None<VALUE>
             </ITEM>
        <ITEM_LIST>
    </RESPONSE>
</SIMPLE_RETURN>
```

KnowledgeBase API

Qualys' Sofware-as-a-Service (SaaS) technology includes its KnowledgeBase, with the industry's largest number of vulnerability signatures, that is continuously updated by Qualys' Research and Development team. Qualys is fully dedicated to providing the most accurate security audits in the industry. Each day new and updated signatures are tested in Qualys' own vulnerability labs and then published, making them available to Qualys customers.

The KnowledgeBase API V2 (**`api/2.0/fo/knowledge_base/vuln/?action=list`**) allows API users to download a list of vulnerabilities from Qualys' KnowledgeBase. This V2 API offers numerous benefits over the knowledgebase_download.php V1 function. Several input parameters grant users control over which vulnerabilities to download and the amount of detail to download, and the XML output provides a rich information source for each vulnerability. The GET or POST access method may be used to make an API request.

Authorized Qualys users have the ability to download vulnerability data using the KnowledgeBase API V2. Please contact Qualys Support or your sales representative if you would like to obtain authorization for your subscription.

Permissions

User permissions for the KnolwedgeBase API V2 are described below. Note: Your subscription must be granted permission to run this function. Please contact Qualys Support or your sales representative to receive this authorization.

User Role	Permissions
Manager, Unit Manager, Scanner, Reader	Download vulnerability data from the KnowledgeBase.
Auditor	No permission to download vulnerability data from the KnowledgeBase.

Parameters

The input parameters for the KnowledgeBase API V2 are described below. Several optional input parameters may be specified. When unspecified, the XML output includes all vulnerabilities in the KnowledgeBase, showing basic details for each vulnerability. Several optional parameters allow you specify filters. When filter parameters are specified, these parameters are ANDed by the service to filter the data from the output.

Parameter	Description
action=list	(Required) A flag used to request the download of vulnerability data from the KnowledgeBase.

Parameter	Description
echo_request={**0** | 1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When unspecified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
details={**Basic** | All | None}	(Optional) Show the requested amount of information for each vulnerability in the XML output. A valid value is: Basic (default), All, or None. Basic includes basic elements plus CVSS Base and Temporal scores. All includes all vulnerability details, including the Basic details.
ids={value}	(Optional) Used to filter the XML output to include only vulnerabilities that have QID numbers matching the QID numbers you specify.
id_min={value}	(Optional) Used to filter the XML output to show only vulnerabilities that have a QID number greater than or equal to a QID number you specify.
id_max={value}	(Optional) Used to filter the XML output to show only vulnerabilities that have a QID number less than or equal to a QID number you specify.
is_patchable={0 | 1}	(Optional) Used to filter the XML output to show only vulnerabilities that are patchable or not patchable. A vulnerability is considered patchable when a patch exists for it. When 1 is specified, only vulnerabilities that are patchable will be included in the output. When 0 is specified, only vulnerabilities that are not patchable will be included in the output. When unspecified, patchable and unpatchable vulnerabilities will be included in the output.
last_modified_after={date}	(Optional) Used to filter the XML output to show only vulnerabilities last modified after a certain date and time. When specified vulnerabilities last modified by a user or by the service will be shown. The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT).
last_modified_before={date}	(Optional) Used to filter the XML output to show only vulnerabilities last modified before a certain date and time. When specified vulnerabilities last modified by a user or by the service will be shown. The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT).
last_modified_by_user_after={date}	(Optional) Used to filter the XML output to show only vulnerabilities last modified by a user after a certain date and time. The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT).

Parameter	Description
last_modified_by_user_before={date}	(Optional) Used to filter the XML output to show only vulnerabilities last modified by a user before a certain date and time. The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT).
last_modified_by_service_after={date}	(Optional) Used to filter the XML output to show only vulnerabilities last modified by the service after a certain date and time. The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT).
last_modified_by_service_before={date}	(Optional) Used to filter the XML output to show only vulnerabilities last modified by the service before a certain date and time. The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT).
published_after={date}	(Optional) Used to filter the XML output to show only vulnerabilities published after a certain date and time. The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT).
published_before={date}	(Optional) Used to filter the XML output to show only vulnerabilities published before a certain date and time. The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT).
discovery_method={value}	(Optional) Used to filter the XML output to show only vulnerabilities assigned a certain discovery method. A valid value is: Remote, Authenticated, RemoteOnly, AuthenticatedOnly, or RemoteAndAuthenticated. When "Authenticated" is specified, the service shows vulnerabilities that have at least one associated authentication type. Vulnerabilities that have at least one authentication type can be detected in two ways: 1) remotely without using authentication, and 2) using authentication.
discovery_auth_types={value}	(Optional) Used to filter the XML output to show only vulnerabilities having one or more authentication types. A valid value is: Windows, Oracle, Unix or SNMP. Multiple values are entered as a comma-separated list.
show_pci_reasons={**0** | 1}	(Optional) Used to filter the XML output to show reasons for passing or failing PCI compliance (when the CVSS Scoring feature is turned on in the user's subscription). Specify 1 to view the reasons in the XML output. When unspecified, the reasons are not included in the XML output.

Sample API Requests

These sample requests work on Qualys US Platform 1 where the FQDN in the API server URL is qualysapi.qualys.com. Please be sure to replace the FQDN with the proper API server URL for your platform. For a partner platform, use the URL for your @customer platform API server.

Sample 1. Request all vulnerabilities in the KnowledgeBase showing basic details:

```
curl -u "user:password" -H "X-Requested-With: Curl" -X "POST"
-d "action=list"
"https://qualysapi.qualys.com/api/2.0/fo/knowledge_base/vuln/" >
output.txt
```

Sample 2. Request patchable vulnerabilities that have QIDs 1-200 showing all details:

```
curl -u "user:password" -H "X-Requested-With: Curl" -X "POST"
-d "action=list&ids=1-200&is_patchable=1&details=All"
"https://qualysapi.qualys.com/api/2.0/fo/knowledge_base/vuln/" >
output.txt
```

Sample 3. Request vulnerabilites that were last modified by the service after July 20, 2011 and that have the "remote and authenticated" discovery method:

```
curl -u "user:password" -H "X-Requested-With: Curl" -X "POST"
-d "action=list&last_modified_by_service_after=2011-07-20
&discovery_method=RemoteAndAuthenticated"
"https://qualysapi.qualys.com/api/2.0/fo/knowledge_base/vuln/" >
output.txt
```

XML Output

A KnowledgeBase API request returns XML output using the knowledge_base_vuln_list_output.dtd, which can be found at the following URL (where qualysapi.qualys.com is your API server URL):

```
https://qualysapi.qualys.com/api/2.0/fo/knowledge_base/vuln/
knowledge_base_vuln_list_output.dtd
```

The DTD for the KnowledgeBase output is provided in Appendix A.

Share PCI Scan

The Share PCI Scan API v2 (**/api/2.0/fo/scan/pci/**) provides an automated way to share (export) finished PCI scans to PCI Merchant accounts and check the export status. A PCI scan is a vulnerability scan that was run with the option profile "Payment Card Industry (PCI) Options".

Express Lite: This API is available to Express Lite users.

In advance of sharing a PCI scan using the share PCI scan API, the target PCI Merchant account must be already defined as a PCI account link within the API user's Qualys account. Account links can be defined using the Qualys user interface only.

Permissions. Any user with scan permissions (Manager, Unit Manager or Scanner) can share a PCI scan with one of their own PCI Merchant accounts and obtain share status. The user's Qualys account must allow access to the PCI scan and must have a link to the target PCI Merchant account.

Share Restriction. The following share restriction applies to all users. One PCI scan can be shared (exported) to one PCI Merchant subscription one time only, assuming the share request is successful. (Note: If a particular scan has been exported to any PCI account in the same PCI Merchant subscription as your PCI account, the scan can't be exported.) If a share request fails for some reason, it's possible to submit another share request for the same PCI scan and PCI Merchant account.

Share a PCI Scan

The share PCI scan API (**/api/2.0/fo/scan/pci/**) with **action=share** is used to export a finished PCI scan to a selected PCI Merchant account. It's possible to export a PCI scan one time per PCI Merchant account, and the same PCI scan can be exported to multiple PCI Merchant accounts. For a share request, the allowed method is: POST.

The input parameters used to make a request to share a PCI scan are below.

Parameter	Description
action=share	(Required) Specify "share" to share a PCI scan.
echo_request={**0** | 1}	(Optional) Specify 1 to view parameters in the XML output. When unspecified, parameters are not included in the XML output.
scan_ref={value}	(Required) The scan reference of a finished PCI scan. The scan status of this scan must be "Finished".
merchant_username={value}	(Required) The user name of the PCI Merchant account that the PCI scan will be exported to. The API user's Qualys account must have a PCI account link already defined for this target PCI Merchant account.

Sample Share Request

This sample request works on Qualys US Platform 1 where the FQDN in the API server URL is qualysapi.qualys.com. Please be sure to replace the FQDN with the proper API server URL for your platform. For a partner platform, use the URL for your @customer platform API server. The header parameter "X-Requested-With" is also provided as an example.

Sample:

```
curl -s -H 'X-Requested-With: curl demo 2' -D headers.15 -b
'QualysSession=38255848108d68a2feaf9ee664ca78a7; path=/api;
secure' -d
'action=share&merchant_username=manager1@qualys&scan_ref=scan/1281
646610.5720'
'https://qualysapi.qualys.com/api/2.0/fo/scan/pci/'
```

Sample Share Output

Successful Share

When the request to share a PCI scan is successful, the XML output uses the simple return DTD and the message is "Requested share of scan to PCI".

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2012-01-17T00:50:39Z</DATETIME>
    <TEXT>Requested share of scan to PCI</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>scan_ref</KEY>
        <VALUE>scan/1281646610.5720</VALUE>
      </ITEM>
      <ITEM>
        <KEY>merchant_username</KEY>
        <VALUE>manager1@qualys</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

Share Already in Progress or Completed

When the request to share a PCI scan fails, the XML output uses the simple return DTD with the error. If the failure is because sharing is in progress for the PCI Merchant account or the scan has already been shared to the PCI account, the output includes the message "This scan has already been shared with the Merchant account".

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2012-01-04T14:54:01Z</DATETIME>
    <CODE>999</CODE>
    <TEXT>This scan has already been shared with the Merchant
account.</TEXT>
  </RESPONSE>
</SIMPLE_RETURN>
```

Get Share Status

The share PCI scan API (**/api/2.0/fo/scan/pci/**) with **action=status** is used to get the share status of a PCI scan that has already been shared. For a status request, the allowed methods are: POST and GET.

The input parameters used to make a request to get the PCI scan share status are below.

Parameter	Description
action=status	(Required) Specify "status" to return the status of a PCI scan export for a given scan report reference and PCI Merchant account.
echo_request={**0** | 1}	(Optional) Specify 1 to view parameters in the XML output. When unspecified, parameters are not included in the XML output.
scan_ref={value}	(Required) The scan reference of the shared scan that you want to check the export status for.
merchant_username={value}	(Required) The username of the PCI account which the scan was exported to.

Sample Status Request

This sample request works on Qualys US Platform 1 where the FQDN in the API server URL is qualysapi.qualys.com. Please be sure to replace the FQDN with the proper API server URL for your platform. For a partner platform, use the URL for your @customer platform API server. The header parameter "X-Requested-With" is also provided as an example.

Sample:

```
curl -s -H 'X-Requested-With: curl demo 2' -D headers.15 -b
'QualysSession=38255848108d68a2feaf9ee664ca78a7; path=/api;
secure' -d
'action=status&merchant_username=manager1@qualys&scan_ref=scan/128
1646610.5720' 'https://qualysapi.qualys.com/api/2.0/fo/scan/pci/'
```

Sample Status Output

The PCI scan share status XML output uses the PCI scan share status DTD, as shown below.

The XML response for a status requests identifies the share status: Queued (request was received and not started yet), In Progress, Finished (scan was exported to PCI account successfully), or Error.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE PCI_SCAN_SHARE_STATUS SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/scan/pci/pci_scan_share_s
tatus.dtd">
<PCI_SCAN_SHARE_STATUS>
  <RESPONSE>
    <SCAN>
      <MERCHANT_USERNAME>manager1@qualys</MERCHANT_USERNAME>
      <SCAN_REF>scan/1281646610.5720</SCAN_REF>
      <STATUS>Finished</STATUS>
      <LAST_SHARED>1970-01-01T00:33:32Z</LAST_SHARED>
    </SCAN>
  </RESPONSE>
</PCI_SCAN_SHARE_STATUS>
```

PCI Scan Share Status DTD

An API request to get the share status of a PCI scan that has been shared returns XML output using the "pci_scan_share_status.dtd" which can be found at the following URL (where <qualysapi.qualys.com> identifies the Qualys API server URL where your account is located):

```
https://qualysapi.qualys.com/api/2.0/fo/scan/pci/
pci_scan_share_status.dtd
```

The DTD for "pci_scan_share_status.dtd" is provided in Appendix A.

CHAPTER 4

Report API

Report Share is a subscription level feature. When enabled, completed reports can be posted to Report Share where they can be shared with other users. This promotes collaboration on vulnerability issues within the IT infrastructure. Report Share is available through both the Qualys user interface and Qualys API.

Report Share is available to users with Enterprise accounts; it is not available to users with Express accounts. Please contact Qualys Support if you would like information on how to upgrade your account to use Report Share.

This chapter describes how to use the Report Share API, which is built on the API V2 Architecture.

These topics are covered:

- About Report Share
- Report List
- Launch Report
- Launch Scorecard
- Cancel Running Report
- Download Saved Report
- Delete Saved Report

About Report Share

Report Share provides these enhanced reporting features:

- Allows for generation, storage and distribution of reports for large enterprise networks
- Enables API users to automate report generation, retrieval and management
- Report email notification (optional) allows users to review reports upon completion

The Report Share capabilities available in the Report Share API mirror the same capabilities available in the Qualys user interface. Users may opt in to receive report email notifications by editing their account in the Qualys user interface.

The Report Share APIs enable users to programmatically launch reports based on existing report templates in the user's account, as well as to list, manage and download saved reports in the user's account. When making an API request to launch a report, the user must specify the output format (each report type has a list of available output formats). While a report is running, the user can make API requests for a report list to check the percentage complete and when report generation has completed. Once the report is generated and saved to Report Share, the user can make an API request to fetch (download) the report. Like API requests for scans, the user has the option make API requests to cancel a running report and to delete a saved report.

Qualys manages report storage. Reports are available in Report Share storage until they expire 7 days after the report creation date. Each user is allocated an amount of disk space for reports. See the Qualys online help for more information.

The Report Share API allows users to perform the following actions.

Action	Supported Access Method	Description
Launch	POST	Launch a report for a selected report template and format.
List	GET POST	List reports in the user's Report Share storage space. Both in progress and saved reports are included.
Cancel	POST	Cancel a running report in the user's Report Share.
Delete	POST	Delete a report in the user's Report Share.
Fetch	GET POST	Download a report in the user's Report Share.

Report List

The **/api/2.0/fo/report** resource with the parameter **action=list** is used to view a list of reports in the user's Report Share. The report list output includes all report types, including scorecard reports. Authentication is required for each API request. See "Authentication Using the V2 APIs."

The XML output identifies reports in progress and saved reports. This information is provided for each report: report ID, title, name of user who launched the report, the date launched, report format, report size (in megabytes), report status (finished or percentage complete), and number of days before the report will expire.

The GET or POST access method may be used to make a report list request.

User permissions for viewing reports are described below.

User Role	Permissions
Manager	View all reports in subscription.
Auditor	View all policy reports in subscription.
Unit Manager	View reports in user's business unit, including reports launched by the user and reports launched by other users in the same business unit.
Scanner	View reports launched by the user.
Reader	View reports launched by the user.

Parameters

The parameters used to request a report list are described below.

Parameter	Description
action=list	(Required) A flag used to make a request for a report list.
echo_request={**0** | 1}	(Optional) Specifies whether to echo the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
id={value}	(Optional) Specifies a report ID of a report that is saved in the Report Share storage space. When specified, information on the selected report will be included in the XML output.
state={value}	(Optional) Specifies that reports with a certain state will be included in the XML output. By default, all states are included. A valid value is: Submitted (reports launched, not yet started to run), Running (reports in progress), Finished, Canceled, or Errors.

Parameter	Description
user_login={value}	(Optional) Specifies a user login ID. This parameter is used to restrict the XML output to reports launched by the specified user login ID.
expires_before_datetime={date}	(Optional) Specifies the date and time (optional) when reports will expire in the future. Only reports that expire before this date/time will be included in the XML output. The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT), like "2007-07-01" or "2007-01-25T23:12:00Z".

Example

A sample report list API call (GET method) is shown below.

```
curl -H "X-Requested-With: Curl Sample"
-b "QualysSession=71e6cda2a35d2cd404cddaf305ea0208; path=/api;
secure" "https://qualysapi.qualys.com/api/2.0/fo/report/
?action=list"

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE REPORT_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/report/report_list_out
put.dtd">

<REPORT_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2009-10-30T22:32:15Z</DATETIME>
    <REPORT_LIST>
      <REPORT>
        <ID>42703</ID>
        <TITLE><![CDATA[Test now]]></TITLE>
        <TYPE>Scan</TYPE>
        <USER_LOGIN>acme_aa</USER_LOGIN>
        <LAUNCH_DATETIME>2009-10-30T17:59:22Z</LAUNCH_DATETIME>
        <OUTPUT_FORMAT>PDF</OUTPUT_FORMAT>
        <SIZE>129.1 MB</SIZE>
        <STATUS>
          <STATE>Finished</STATE>
        </STATUS>
        <EXPIRATION_DATETIME>2009-11-
06T17:59:24Z</EXPIRATION_DATETIME>
```

```
      </REPORT>
      <REPORT>
        <ID>42700</ID>
        <TYPE>Scorecard</TYPE>
        <USER_LOGIN>acme_ts2</USER_LOGIN>
        <LAUNCH_DATETIME>2009-10-29T22:12:42Z</LAUNCH_DATETIME>
        <OUTPUT_FORMAT>SECURE_PDF</OUTPUT_FORMAT>
        <SIZE>18.1 KB</SIZE>
        <STATUS>
          <STATE>Finished</STATE>
        </STATUS>
        <EXPIRATION_DATETIME>2009-11-
05T22:12:44Z</EXPIRATION_DATETIME>
      </REPORT>
      <REPORT>
        <ID>42699</ID>
        <TYPE>Scorecard</TYPE>
        <USER_LOGIN>quays_ts2</USER_LOGIN>
        <LAUNCH_DATETIME>2009-10-29T21:52:19Z</LAUNCH_DATETIME>
        <OUTPUT_FORMAT>PDF</OUTPUT_FORMAT>
        <SIZE>19.87 KB</SIZE>
        <STATUS>
          <STATE>Finished</STATE>
        </STATUS>
        <EXPIRATION_DATETIME>2009-11-
05T21:52:21Z</EXPIRATION_DATETIME>
      </REPORT>
    </REPORT_LIST>
  </RESPONSE>
</REPORT_LIST_OUTPUT>
```

XML Output

A report list request (parameter **`action=list`**) returns XML output using the DTD "report_list_ouput.dtd" which can be found at the following URL:

```
https://qualysapi.qualys.com/api/2.0/fo/report/
report_list_output.dtd
```

The DTD for the report list XML output is provided in Appendix A.

Launch Report

The **`/api/2.0/fo/report`** resource with the parameter **`action=launch`** is used to launch a report in the user's Report Share. Authentication is required for each API request. See Chapter 2, "Authentication Using the V2 APIs."

When a report is launched with Report Share, the report is run in the background, and the report generation processing does not timeout until the report has completed.

The POST access method must be used to make a request to launch a report.

User permissions for launching reports are described below.

User Role	Permissions
Manager	Launch report for all assets (IPs, domains) in subscription.
Auditor	Launch policy report for all IPs in subscription.
Unit Manager	Launch report for assets in user's business unit.
Scanner	Launch report for assets in user's account.
Reader	Launch report for assets in user's account.

Report Source

When launching a manual scan report, you specify a scan reference to include. Optionally you may specify IPs/ranges to restrict the report to certain IPs.

When launching other reports, you can specify asset groups and IPs/ranges to include. Hosts may be specified as follows:

Host Entry	Example
Single IP	123.123.123.1
Multiple IPs	123.123.123.1,123.123.123.4,123.123.123.5
IP Ranges	
IP Range	123.123.123.1-123.123.123.8
IP Range and Single IPs	10.10.10.1-10.10.10.100,64.41.134.60,64.41.134.66

When specifying hosts, these syntax elements may be used:

- Multiple IPs/ranges are separated by a comma (,).
- For an IP range, use a dash (-) to separate the first and last IP.

All Report Requests

Input parameter for all API reporting requests are described below. See the following sections for input parameters per report type.

Parameter	Description
action=launch	(Required) A flag used to make a request to launch a report.
echo_request={**0** | 1}	(Optional) Specifies whether to echo the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
template_id={value}	(Required) The template ID of the report you want to launch. Use the **report_template_list.php** function to find the appropriate template ID.
report_title=[value}	(Optional) A user-defined report title. The title may have a maximum of 128 characters. For a PCI compliance report, the report title is provided by Qualys and cannot be changed.
output_format={value}	(Required) One output format may be specified. When **output_format=pdf** is specified, the Secure PDF Distribution may be used. See "Launch Report Sample." For a map report, a valid value is: pdf, html (a zip file), mht, xml, or csv For a scan report, a valid value is: pdf, html (a zip file), mht, xml, csv, or docx For a remediation report, a valid value is: pdf, html (a zip file), mht, or csv For a compliance report (not PCI), a valid value is: pdf, html (a zip file), or mht For a PCI compliance report, a valid value is: pdf or html (a zip file) For a compliance policy report, a valid value is: pdf, html (a zip file), mht, xml, or csv For a Qualys patch report, a valid value is: pdf, online, or csv
hide_header={**0** | 1}	(Valid for CSV format report only). Specify **hide_header=1** to omit the header information from the report. By default this information is included.

Launch Report Sample

A sample launch report API call (POST method) is shown below.

```
curl -H "X-Requested-With: Curl Sample"
-d "action=launch&template_id=55469&output_format=pdf"
-b "QualysSession=71e6cda2a35d2cd404cddaf305ea0208; path=/api;
secure" "https://qualysapi.qualys.com/api/2.0/fo/report/"

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE GENERIC SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2013-06-20T21:45:23Z</DATETIME>
    <TEXT>New report launched</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>1665</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

Report Distribution

Notify Users

Notify users when the report is complete using the parameter "recipient_group_id". .

Parameter	Description
recipient_group_id={value}	(Optional) Specify users who will receive the email notification when the report is complete (i.e. supply a distribution group ID). Where do I find this ID? Log in to your Qualys account, go to Users > Distribution Groups and select Info for a group in the list.

Sample request .

```
curl -H "X-Requested-With: Curl Sample" -d
"action=launch&template_id=55469&output_format=pdf&recipient_group_id=123
456"
-b "QualysSession=71e6cda2a35d2cd404cddaf305ea0208; path=/api;
secure" "https://qualysapi.qualys.com/api/2.0/fo/report/"
```

Secure PDF Distribution

Once this feature is enabled for the subscription, Managers and Unit Managers can create encrypted PDF reports and securely distribute them to users via email. Want to learn more? See "Securely Distribute PDF Reports" in the Qualys online help .

Use these parameters when you launch a PDF report (**`output_format=pdf`**.

Parameter	Description
pdf_password={value}	(Optional; Required for secure PDF distribution) The password to be used for encryption. Requirements: - the password must have a minimum of 8 characters (ascii), and a maximum of 32 characters - the password must contain alpha and numeric characters - the password cannot match the password for the user's Qualys account. - the password must follow the password security guidelines defined for your subscription (log in and go to Subscription Setup—>Security Options).
recipient_group={value}	(Optional; Optional for secure PDF distribution) The report recipients in the form of one or more distribution groups, as defined using the Qualys UI. Multiple distribution groups are comma separated. A maximum of 50 distribution groups may be entered.

Map Report

Input parameters for an API request that launches a map report are described below. Only valid parameters for this type of request are listed.

Parameter	Description
report_type=Map	(Optional) Specifies the template type of the report. For a map report, specify Map.
output_format={value}	(Required) Specifies the output format of the report. One output format may be specified. For a map report, a valid value is: pdf, html (a zip file), mht, xml, or csv.
hide_header={**`0`**|1}	(Valid for CSV format report only). Specify **`hide_header=1`** to omit the header information from the report. By default this information is included.
domain={value}	(Required for map report) Specifies the target domain for the map report. Include the domain name only; do not enter "www." at the start of the domain name. When the special "none" domain is specified as a parameter value, the **`ip_restriction`** parameter is required.

Parameter	Description
ip_restriction={value}	(Optional for map report) For a map report, specifies certain IPs/ranges to include in the report. This parameter is required when the **domain** parameter is specified with the value "none" (for the special "none" domain). Multiple IPs and/or ranges are comma separated.
report_refs={value}	(Required for map report) For a map report, specifies the map references (1 or 2) to include. A map reference starts with the string "map/" followed by a reference ID number. When two map references are given, the report compares map results. Two map references are comma separated.

Scan Report: Scan Based Findings (manual)

Input parameters for an API request that launches a scan report including scan based findings are described below. Only valid parameters for this type of request are listed.

Parameter	Description
report_type=Scan	(Optional) Specifies the template type of the report. For a scan report, specify Scan.
output_format={value}	(Required) Specifies the output format of the report. One output format may be specified. For a scan report, a valid value is: pdf, html (a zip file), mht, xml, or csv.
hide_header={**0**|1}	(Valid for CSV format report only). Specify **hide_header=1** to omit the header information from the report. By default this information is included.
report_refs={value}	(Required for Manual scan report) For a Manual scan report, this parameter specifies the scan references to include. A scan reference starts with the string "scan/" followed by a reference ID number. Multiple scan references are comma separated.
ip_restriction={value}	(Optional for Manual scan report) For a scan report, the report content will be restricted to the specified IPs/ranges. Multiple IPs and/or ranges are comma separated.

Scan Report: Host Based Findings (automatic)

Input parameters for an API request that launches an scan report including host based findings are below. Only valid parameters for this type of request are listed.

Parameter	Description
report_type=Scan	(Optional) Specifies the template type of the report. For a scan report, specify Scan.
output_format={value}	(Required) Specifies the output format of the report. One output format may be specified. For a scan report, a valid value is: pdf, html (a zip file), mht, xml, or csv.
hide_header={**0**|1}	(Valid for CSV format only). Specify **hide_header=1** to omit the header information from the report. By default this information is included.
ips={value}	(Optional) Specify IPs/ranges to change (override) the report target, as defined in the scan report template. Multiple IPs/ranges are comma separated. When specified, hosts defined in the report template are not included in the report. See also "Using Asset Tags."
asset_group_ids={value}	(Optional) Specify asset group IDs to change (override) the report target, as defined in the scan report template. When specified, hosts defined in the report template are not included in the report. Looking for asset group IDs? Use the **asset_group_list.php** function (see the API v1 User Guide).
ips_network_id={value}	(Optional, and valid only when the Network Support feature is enabled for the user's account) The ID of a network that is used to restrict the report's target to the IPs/ranges specified in the"ips" parameter. Set to a custom network ID (note this does not filter IPs/ranges specified in "asset_group_ids"). Or set to "0" (the default) for the Global Default Network - this is used to report on hosts outside of your custom networks.

Patch Report

Input parameters for an API request that launches a Qualys Patch Report are described below. Only valid parameters for this type of request are listed. By default, an API request to launch a patch report uses the report source defined in the report template.

Parameter	Description
output_format={value}	(Required) Specifies the output format of the report. One output format may be specified. For a patch, a valid value is: pdf, online, or csv.
hide_header={**0** | 1}	(Valid for CSV format report only). Specify **hide_header=1** to omit the header information from the report. By default this information is included.
ips={value}	(Optional for patch report) Specify IPs/ranges to change (override) the report target, as defined in the patch report template. Multiple IPs/ranges are comma separated. When specified, hosts defined in the report template are not included in the report. See also "Using Asset Tags."
asset_group_ids={value}	(Optional for patch report) Specify IPs/ranges to change (override) the report target, as defined in the patch report template. Multiple asset group IDs are comma separated. When specified, hosts defined in the report template are not included in the report. Looking for asset group IDs? Use the **asset_group_list.php** function (see the API v1 User Guide).

Remediation Report

Input parameters for an API request that launches a remediation report are described below. Only valid parameters for this type of request are listed.

The parameters **ips** and/or **asset_group_ids** are used to specify hosts to include in the report. One or both parameters must be specified. When both parameters are specified, all hosts specified by the parameter values are combined (ANDed) and included in the report.

By default, tickets assigned to the current user are included in the report. Using the optional **assignee_type** parameter, you can include all tickets in the user account.

Parameter	Description
report_type=Remediation	(Optional) Specifies the template type of the report. For a remediation report, specify Remediation.
output_format={value}	(Required) Specifies the output format of the report. One output format may be specified. For a remediation report, a valid value is: pdf, html (a zip file), mht, or csv.

Parameter	Description
hide_header={**0** | 1}	(Valid for CSV format report only). Specify **hide_header=1** to omit the header information from the report. By default this information is included.
ips={value}	(Optional for remediation report) Specify IPs/ranges you want to include in the report. Multiple IPs and/or ranges are comma separated.
asset_group_ids={value}	(Optional for remediation report) Specify asset group IDs that identify hosts you want to include in the report. Multiple asset group IDs are comma separated. Looking for asset group IDs? Use the **asset_group_list.php** function (in the API v1 User Guide).
assignee_type={**User** | All}	(Optional for remediation report) Specifies whether the report will include tickets assigned to the current user, or all tickets in the user account. By default tickets assigned to the current user are included. Valid values are: User (default) or All.

Compliance Report

Input parameters are described below for an API request that launches a compliance report based on vulnerability management data. Only valid parameters for this type of request are listed. Compliance reports are: Qualys Top 20 Report, SANS Top 20 Report, Qualys PCI Executive Report, and Qualys PCI Technical Report.

The parameters **ips** and/or **asset_group_ids** are used to specify hosts to include in the Qualys Top 10 Report and SANS Top 20 Report. One or both parameters must be specified. When both parameters are specified, all hosts specified by the parameter values are combined (ANDed) and included in the report.

Parameter	Description
report_type=Compliance	(Optional) Specifies the template type of the report. For a compliance report, specify Compliance.
output_format={value}	(Required) Specifies the output format of the report. One output format may be specified. For a compliance report other than a PCI report, a valid value is: pdf, html (a zip file), or mht. For a PCI report, a valid value is: pdf or html (a zip file).

Parameter	Description
ips={value}	(Optional for compliance report) For a compliance report (except a PCI report), specify the IPs/ranges you want to include in the report. Multiple IPs and/or ranges are comma separated. This parameter or the **asset_group_ids** parameter must be specified for these reports: Qualys Top 20 Report, 2008 SANS Top 20 Report.
	Optional: Qualys Top 20 Report, SANS Top 20 Report
	Invalid: PCI Executive Report, PCI Technical Report
asset_group_ids={value}	(Optional for compliance report) For a compliance report (except a PCI report), specify asset groups IDs which identify hosts to include in the report. Multiple asset group IDs are comma separated. Looking for asset group IDs? Use the **asset_group_list.php** function (in the API v1 User Guide). This parameter or the **ips** parameter must be specified for these reports: Qualys Top 20 Report, SANS Top 20 Report.
	Optional: Qualys Top 20 Report, SANS Top 20 Report
	Invalid: PCI Executive Report, PCI Technical Report
report_refs={value}	(Required for PCI compliance report) For a PCI compliance report, either the technical or executive report, this parameter specifies the scan reference to include. A scan reference starts with the string "scan/" followed by a reference ID number. The scan reference must be for a scan that was run using the PCI Options profile. Only one scan reference may be specified.
	Required: PCI Executive Report, PCI Technical Report
	Invalid: Qualys Top 20 Report, SANS Top 20 Report

Compliance Policy Report

Input parameters for an API request that launches a compliance policy report are described below. Only valid parameters for this type of request are listed.

Parameter	Description
report_type=Policy	(Optional) Specifies the template type of the report. For a compliance policy report, specify Policy.
output_format={value}	(Required) Specifies the output format of the report. One output format may be specified. For a compliance report other than a PCI report, a valid value is: pdf, html (a zip file), mht, xml, or csv.

Parameter	Description
hide_header={**0**|1}	(Valid for CSV format report only). Specify **hide_header=1** to omit the header information from the report. By default this information is included.
policy_id={value}	(Required) Specifies the policy to run the report on. A valid policy ID must be entered.
asset_group_ids={value}	(Optional) Specify asset group IDS if you want to include only certain asset groups in your report. These asset groups must be assigned to the policy you are reporting on. Multiple asset group IDs are comma separated. Looking for asset group IDs? Use the **asset_group_list.php** function (in the API v1 User Guide).
	Want to limit your report to certain IP addresses? You can select asset group IDs (**asset_group_ids**), IP addresses/ranges (**ips**) or asset tags. See "Using Asset Tags."
ips={value}	(Optional) Specify IPs/ranges if you want to include only certain IP addresses in your report. These IPs must be assigned to the policy you're reporting on. Multiple entries are comma separated.
	Want to limit your report to certain IP addresses? You can select asset group IDs (**asset_group_ids**), IP addresses/ranges (**ips**) or asset tags. See "Using Asset Tags."
host_id={value}	(Optional) In the policy report output, show only results for a single host instance. Specify the ID for the host to include in the report. A valid host ID must be entered. This parameter must be specified with: **instance_string**.
instance_string={value}	(Optional) Specifies a single instance on the selected host. The instance string may be "os" or a string like "oracle10:1:1521:ora10204u". Use the "Compliance Posture Information" API v2 (with the endpoint/api/2.0/fo/compliance/posture/info) to find the appropriate instance string. This parameter must be specified with: **host_id**.

Using Asset Tags

It's possible to select asset tags for both vulnerability and compliance reports. Use the following tag parameters to launch your report using asset tags.

Parameter	Description
use_tags={**0** | 1}	(Optional) Specify "1" when your report target will include asset tags. Specify "0" (the default) when your report target will include IP addresses/ranges and/or asset groups. When not specified, use_tags=0 is used.
tag_include_selector={all | **any**}	(Optional) Select "any" (the default) to include hosts that match at least one of the selected tags. Select "all" to include hosts that match all of the selected tags.
	tag_include_selector is valid only when use_tags=1 is specified.
tag_exclude_selector={all | **any**}	(Optional) Select "any" (the default) to exclude hosts that match at least one of the selected tags. Select "all" to exclude hosts that match all of the selected tags.
	tag_exclude_selector is valid only when use_tags=1 is specified.
tag_set_by={**id** | name}	(Optional) Specify "id" (the default) to select a tag set by providing tag IDs. Specify "name" to select a tag set by providing tag names.
	tag_set_by is valid only when use_tags=1 is specified.
tag_set_include={value}	(Optional) Specify a tag set to include. Hosts that match these tags will be included. You identify the tag set by providing tag name or IDs. Multiple entries are comma separated.
	tag_set_include is valid only when use_tags=1 is specified.
tag_set_exclude={value}	(Optional) Specify a tag set to exclude. Hosts that match these tags will be excluded. You identify the tag set by providing tag name or IDs. Multiple entries are comma separated.
	tag_set_exclude is valid only when use_tags=1 is specified.

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST"
-d
"action=launch&template_id=55469&report_title=My+Windows+Report&ou
tput_format=pdf&use_tags=1&tag_set_by=name&tag_set_include=Windows
" "https://qualysapi.qualys.com/api/2.0/fo/report/"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE GENERIC SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
    <RESPONSE>
        <DATETIME>2014-02-20T21:45:23Z</DATETIME>
        <TEXT>New report launched</TEXT>
        <ITEM_LIST>
            <ITEM>
                <KEY>ID</KEY>
                <VALUE>1665</VALUE>
            </ITEM>
        </ITEM_LIST>
    </RESPONSE>
</SIMPLE_RETURN>
```

Report Template List

The **report_template_list.php** function provides a list of available report templates, including template titles and IDs, in the user account. The report list includes templates for all report types.

To retrieve a list of report templates, use this URL:

```
https://qualysapi.qualys.com/msp/report_template_list.php
```

The DTD for the XML document returned from **report_template_list.php** can be found at the following URL:

```
https://qualysapi.qualys.com/report_template_list.dtd
```

Sample report template list output is shown below:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE REPORT_TEMPLATE_LIST SYSTEM
"https://qualysapi.qualys.com/report_template_list.dtd">
<REPORT_TEMPLATE_LIST>
  <REPORT_TEMPLATE>
    <ID>235288</ID>
    <TYPE>Auto</TYPE>
    <TEMPLATE_TYPE>Scan</TEMPLATE_TYPE>
    <TITLE><![CDATA[Windows Authentication QIDs]]></TITLE>
    <USER>
```

```
      <LOGIN><![CDATA[acme_jk]]></LOGIN>
      <FIRSTNAME><![CDATA[Jason]]></FIRSTNAME>
      <LASTNAME><![CDATA[Kim]]></LASTNAME>
    </USER>
    <LAST_UPDATE>2014-02-12T18:09:10Z</LAST_UPDATE>
    <GLOBAL>0</GLOBAL>
  </REPORT_TEMPLATE>
  <REPORT_TEMPLATE>
    <ID>235164</ID>
    <TYPE>Auto</TYPE>
    <TEMPLATE_TYPE>Policy</TEMPLATE_TYPE>
    <TITLE><![CDATA[My Policy Report Template]]></TITLE>
    <USER>
      <LOGIN><![CDATA[acme_vs]]></LOGIN>
      <FIRSTNAME><![CDATA[Victor]]></FIRSTNAME>
      <LASTNAME><![CDATA[Smith]]></LASTNAME>
    </USER>
    <LAST_UPDATE>2013-12-09T22:47:58Z</LAST_UPDATE>
    <GLOBAL>0</GLOBAL>
  </REPORT_TEMPLATE>
  <REPORT_TEMPLATE>
    <ID>232556</ID>
    <TYPE>Auto</TYPE>
    <TEMPLATE_TYPE>Scan</TEMPLATE_TYPE>
    <TITLE><![CDATA[Executive Report]]></TITLE>
    <USER>
      <LOGIN><![CDATA[acme_jk]]></LOGIN>
      <FIRSTNAME><![CDATA[Jason]]></FIRSTNAME>
      <LASTNAME><![CDATA[Kim]]></LASTNAME>
    </USER>
    <LAST_UPDATE>2013-11-11T17:11:55Z</LAST_UPDATE>
    <GLOBAL>1</GLOBAL>
  </REPORT_TEMPLATE>
  <REPORT_TEMPLATE>
    <ID>232557</ID>
    <TYPE>Auto</TYPE>
    <TEMPLATE_TYPE>Scan</TEMPLATE_TYPE>
    <TITLE><![CDATA[Technical Report]]></TITLE>
    <USER>
      <LOGIN><![CDATA[acme_jk]]></LOGIN>
      <FIRSTNAME><![CDATA[Jason]]></FIRSTNAME>
```

```
        <LASTNAME><![CDATA[Kim]]></LASTNAME>
  ...
</REPORT_TEMPLATE_LIST>
```

Each <REPORT_TEMPLATE> element identifies template properties, including the report template ID, template type and title, in the sub-elements described below.

Element	Description
<ID>	The template ID number.
<TYPE>	The template type: Auto (for automatic) or Manual.
<TEMPLATE_TYPE>	The report template type: Scan (for a scan report template) Map (for a map report template) Remediation (for a remediation report template) Compliance (for a compliance report template) Policy (for a compliance policy report template) Patch (for a patch report template)
<TITLE>	The template title, as defined in the Qualys user interface.
<USER>	The template owner, identified by login, first name and last name. For a system template, the login "system" is reported.
<LAST_UPDATE>	The most recent date and time when the template was updated.
<GLOBAL>	For a global template, the value 1 appears. For a non global template, the value 0 appears.

Sample report template list output is shown below:

```
<!DOCTYPE REPORT_TEMPLATE_LIST SYSTEM
"https://qualysapi.qualys.com/report_template_list.dtd">

<REPORT_TEMPLATE_LIST>
  <REPORT_TEMPLATE>
    <ID>12399</ID>
    <TYPE>Auto</TYPE>
    <TEMPLATE_TYPE>Scan</TEMPLATE_TYPE>
    <TITLE><![CDATA[Executive Report]]></TITLE>
    <USER>
      <LOGIN><![CDATA[acme_ab12]]></LOGIN>
      <FIRSTNAME><![CDATA[Alice]]></FIRSTNAME>
      <LASTNAME><![CDATA[Bucher]]></LASTNAME>
    </USER>
    <LAST_UPDATE>2006-12-20T03:21:54Z</LAST_UPDATE>
```

```
  <GLOBAL>1</GLOBAL>
</REPORT_TEMPLATE>
<REPORT_TEMPLATE>
  <ID>12400</ID>
  <TYPE>Auto</TYPE>
  <TEMPLATE_TYPE>Scan</TEMPLATE_TYPE>
  <TITLE><![CDATA[Technical Report]]></TITLE>
  <USER>
    <LOGIN><![CDATA[acme_ab12]]></LOGIN>
    <FIRSTNAME><![CDATA[Alice]]></FIRSTNAME>
    <LASTNAME><![CDATA[Bucher]]></LASTNAME>
  </USER>
  <LAST_UPDATE>2006-12-22T23:11:24Z</LAST_UPDATE>
  <GLOBAL>1</GLOBAL>
</REPORT_TEMPLATE>
<REPORT_TEMPLATE>
  <ID>12403</ID>
  <TYPE>Auto</TYPE>
  <TEMPLATE_TYPE>Compliance</TEMPLATE_TYPE>
  <TITLE><![CDATA[SANS Top 20 Report]]></TITLE>
  <USER>
    <LOGIN><![CDATA[System]]></LOGIN>
    <FIRSTNAME><![CDATA[System]]></FIRSTNAME>
    <LASTNAME><![CDATA[System]]></LASTNAME>
  </USER>
  <LAST_UPDATE>2004-08-09T18:07:17Z</LAST_UPDATE>
  <GLOBAL>1</GLOBAL>
</REPORT_TEMPLATE>
<REPORT_TEMPLATE>
  <ID>12404</ID>
  <TYPE>Auto</TYPE>
  <TEMPLATE_TYPE>Compliance</TEMPLATE_TYPE>
  <TITLE><![CDATA[Qualys Top 20 Report]]></TITLE>
  <USER>
    <LOGIN><![CDATA[System]]></LOGIN>
    <FIRSTNAME><![CDATA[System]]></FIRSTNAME>
    <LASTNAME><![CDATA[System]]></LASTNAME>
  </USER>
  <LAST_UPDATE>2004-08-09T18:07:17Z</LAST_UPDATE>
  <GLOBAL>1</GLOBAL>
</REPORT_TEMPLATE>
```

```
  <REPORT_TEMPLATE>
    <ID>44154</ID>
    <TYPE>N/A</TYPE>
    <TEMPLATE_TYPE>Remediation</TEMPLATE_TYPE>
    <TITLE><![CDATA[Executive Remediation Report]]></TITLE>
    <USER>
      <LOGIN><![CDATA[System]]></LOGIN>
      <FIRSTNAME><![CDATA[System]]></FIRSTNAME>
      <LASTNAME><![CDATA[System]]></LASTNAME>
    </USER>
    <LAST_UPDATE>2006-11-04T02:37:00Z</LAST_UPDATE>
    <GLOBAL>1</GLOBAL>
  </REPORT_TEMPLATE>
  <REPORT_TEMPLATE>
    <ID>44155</ID>
    <TYPE>N/A</TYPE>
    <TEMPLATE_TYPE>Remediation</TEMPLATE_TYPE>
    <TITLE><![CDATA[Tickets per Vulnerability]]></TITLE>
    <USER>
      <LOGIN><![CDATA[System]]></LOGIN>
      <FIRSTNAME><![CDATA[System]]></FIRSTNAME>
      <LASTNAME><![CDATA[System]]></LASTNAME>
    </USER>
    <LAST_UPDATE>2006-11-04T02:37:00Z</LAST_UPDATE>
    <GLOBAL>1</GLOBAL>
  </REPORT_TEMPLATE>
...
  </REPORT_TEMPLATE>
</REPORT_TEMPLATE_LIST>
```

XML Output

A launch report request returns XML output using the DTD “simple_return.dtd”, which can be found at the following URL:

```
https://qualysapi.qualys.com/api/2.0/simple_return.dtd
```

The DTD for the simple return XML output is provided in Appendix A.

Launch Scorecard

The **`/api/2.0/fo/report/scorecard`** resource with the parameter **`action=launch`** is used to launch a vulnerability scorecard report in the user's Report Share. It is not possible to launch any compliance scorecard reports or WAS scorecard reports using this API at this time. Authentication is required for each API request. See Chapter 2, "Authentication Using the V2 APIs."

Service-provided scorecards and user-defined scorecards may be launched. The service-provided scorecards are:

- Asset Group Vulnerability Report. Identifies vulnerabilities with severity levels 3-5
- Ignored Vulnerabilities Report. Identifies vulnerabilities that are currently ignored
- Most Prevalent Vulnerabilities Report. Identifies vulnerabilities with the highest number of detected instances
- Most Vulnerable Hosts Report. Identifies hosts with the highest number of critical vulnerabilities
- Patch Report. Identifies hosts with missing patches and software

When a scorecard report is launched with Report Share, the report is run in the background, and the report generation processing does not timeout until the report has completed.

The POST access method must be used to make a request to launch a scorecard report.

User permissions for launching scorecard reports are described below.

User Role	Permissions
Manager	Launch scorecard report including all IP addresses in subscription.
Auditor	No permissions to launch scorecard reports.
Unit Manager	Launch scorecard report including IP addresses in user's business unit.
Scanner	Launch scorecard report including IP addresses in user's account.
Reader	Launch scorecard report including IP addresses in user's account.

Parameters

The parameters used to launch a scorecard report are below. Parameters are classified as Required, Optional or Conditional (meaning they may be required depending on other parameters specified for the request and user account settings).

Parameter	Description
action=launch	(Required) A flag used to make a request to launch a scorecard report.
echo_request={**0**|1}	(Optional) Specifies whether to echo the request's input parameters (names and values) in the XML output. When unspecified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
name={value}	(Required) Specifies the scorecard name for the vulnerability scorecard report that you want to launch with Report Share. This name corresponds to a service-provided scorecard or a user-created scorecard. For a service-provided scorecard, specify one of these names: Asset Group Vulnerability Report Ignored Vulnerabilities Report Most Prevalent Vulnerabilities Report Most Vulnerable Hosts Report Patch Report
report_title=[value}	(Optional) Specifies a user-defined report title. The title may have a maximum of 128 characters. When unspecified, the report title will be the scorecard name.
output_format={value}	(Required) Specifies the output format of the report. One output format may be specified. A valid value is: pdf, html (a zip file), mht, xml, or csv. When **output_format=pdf** is specified, the Secure PDF Distribution may be used. See "Launch Report Sample."
hide_header={**0**|1}	(Valid for CSV format report only). Specify **hide_header=1** to omit the header information from the report. By default this information is included.

Parameter	Description
pdf_password={value}	(Conditional for Manager or Unit Manager; not valid for other users) The password to be used for encryption. The password may have a maximum of 32 characters (ascii). The password cannot match the password for the user's Qualys login account. The password must follow the password security guidelines defined for the user's subscription. See "Launch Report Sample." Conditions: a) The **pdf_password** parameter can only be specified by a Manager or Unit Manager. b) The **pdf_password** parameter can only be specified when Report Share is enabled for your subscription and the option "Enable Secure PDF Distribution" is selected (Setup—>Report Share).
recipient_group={value}	(Conditional for Manager or Unit Manager; not valid for other users) The report recipients in the form of one or more distribution groups, as defined in your Qualys account. Each distribution group identifies a list of users who will receive the secure PDF report. Multiple distribution groups are comma separated. A maximum of 50 distribution groups may be entered. See "Launch Report Sample." Conditions: a) The **recipient_group** parameter can only be specified when the **pdf_password** parameter is also specified. b) The **recipient_group** parameter can only be specified by a Manager or Unit Manager. c) The **recipient_group** parameter can only be specified when Report Share is enabled for your subscription and the option "Enable Secure PDF Distribution" is selected (Setup—>Report Share).
recipient_group_id={value}	(Optional) Specify users who will receive the email notification when the report is complete (i.e. supply a distribution group ID). Where do I find this ID? Log in to your Qualys account, go to Users > Distribution Groups and select Info for a group in the list.

Parameter	Description
source={value}	(Conditional) The source asset groups for the report. Specify **asset_groups** to select asset groups. Specify **business_unit** to select all the asset groups in a business unit. For a user scorecard, this parameter is optional. When unspecified, the source selection set in the scorecard attributes (as defined in your Qualys account) is used. Conditions: a) The **source** parameter is required for a service-provided scorecard. b) For a user scorecard, the source selection specified in the **source** parameter replaces an existing source selection set in the scorecard attributes (as defined in your Qualys account). If you set this parameter to **asset_groups**, you must specify one of these parameters: **asset_groups** or **all_asset_groups**. If you set this parameter to **business_unit** then you must specify one or more of these parameters: **business_unit**, **division**, **function** and/or **location**.
asset_groups={value}	(Conditional) The titles of asset groups to be used as source asset groups for the scorecard report. One or more asset group titles in your account may be specified. Multiple asset group titles are comma separated. Conditions: a) The **asset_groups** parameter can only be specified when **source=asset_groups**. b) These parameters cannot be specified for the same API request: **asset_groups** and **all_asset_groups**.
all_asset_groups={1}	(Conditional) Set to 1 to select all asset groups available in your account as the source asset groups for the scorecard report. Conditions: a) The **asset_groups** parameter can only be specified when **source=asset_groups**. b) These parameters cannot be specified for the same API request: **asset_groups** and **all_asset_groups**.

Parameter	Description
business_unit={value}	(Conditional for a Manager; not valid for other users) The title of a business unit containing the source asset groups for the scorecard report. All asset groups in the business unit will be included in the report source. You may enter the title of a business unit in your account that was created by a Manager user, or you may enter "Unassigned" for the unassigned business unit. For a user scorecard, the business unit replaces an existing business unit set in the scorecard attributes (as defined in your Qualys account). If an empty value is set (**`business_unit=`**), the existing business unit in the scorecard attributes is not included in the scorecard parameters submitted with the API request. Conditions: a) When **`source=business_unit`**, one or more of these parameters must be specified: **`business_unit`**, **`division`**, **`function`** and/or **`location`**. b) The **`business_unit`** parameter can only be specified by a Manager.
division={value}	(Conditional) A business info tag identifying a division that asset group(s) belong to. The tag must be defined for an asset group in your account. When specified, only asset groups with this tag are included in the scorecard report source. For a user scorecard, the division tag replaces an existing tag set in the scorecard attributes (as defined in your Qualys account). If an empty value is set (**`division=`**), the existing division tag in the scorecard attributes is not included in the scorecard parameters submitted with the API request. Conditions: a) When **`source=business_unit`**, one or more of these parameters must be specified: **`business_unit`**, **`division`**, **`function`** and/or **`location`**. b) The **`division`** parameter can only be specified when **`source=business_unit`**.

Parameter	Description
function={value}	(Conditional) A business info tag identifying a business function for asset group(s). The tag must be defined for an asset group in your account. When specified, only asset groups with this tag are included in the scorecard report source. For a user scorecard, the function tag replaces an existing function tag set in the scorecard attributes (as defined in your Qualys account). If an empty value is set (**function=**), the existing function tag in the scorecard attributes is not included in the scorecard parameters submitted with the API request. Conditions: a) When **source=business_unit**, one or more of these parameters must be specified: **business_unit**, **division**, **function** and/or **location**. b) The **function** parameter can only be specified when **source=business_unit**.
location={value}	(Conditional) A business info tag identifying a location where asset group(s) are located. The tag must be defined for an asset group in your account. When specified, only asset groups with this tag are included in the scorecard report source. For a user scorecard, the location tag replaces an existing location tag set in the scorecard attributes (as defined in your Qualys account). If an empty value is set (**location=**), the existing location tag in the scorecard attributes is not included in the scorecard parameters submitted with the API request. Conditions: a) When **source=business_unit**, one or more of these parameters must be specified: **business_unit**, **division**, **function** and/or **location**. b) The **location** parameter can only be specified when **source=business_unit**.

Parameter	Description
patch_qids={value}	(Conditional for Patch Report scorecard; not valid for other scorecards) Up to 10 QIDs for vulnerabilities or potential vulnerabilities with available patches. Multiple QIDs are comma separated. When the QIDs are detected on a host this means the host does not have the patches installed and it will be reported in the scorecard output. For a user-defined Patch Report, the patch QIDs list replaces the patch QIDs list set in the scorecard attributes (as defined in your Qualys account). If an empty value is set (**patch_qids=**), the existing patches QIDs list in the scorecard attributes is not included in the scorecard parameters submitted with the API request. Conditions: a) The **patch_qids** parameter may be specified only for a Patch Report. b) For a Patch Report, **patch_qids** or **missing_qids** must be specified. Both parameters may be specified together.
missing_qids={value}	(Conditional for Patch Report scorecard; not valid for other scorecards) One or two QIDs for missing software. Two QIDs are comma separated. Typically missing software QIDs are information gathered checks. When the QIDs are not detected on a host this means the host is missing software and it will be reported in the scorecard output. For a user-defined Patch Report, the missing QIDs list replaces the missing QIDs list set in the scorecard attributes (as defined in your Qualys account). If an empty value is set (**missing_qids=**), the existing missing QIDs list in the scorecard attributes is not included in the scorecard parameters submitted with the API request. Conditions: a) The **missing_qids** parameter may be specified only for a Patch Report. b) For a Patch Report, **patch_qids** or **missing_qids** must be specified. Both parameters may be specified together.

XML Output

A launch report request returns XML output using the DTD “simple_return.dtd”, which can be found at the following URL:

```
https://qualysapi.qualys.com/api/2.0/simple_return.dtd
```

The DTD for the simple return XML output is provided in Appendix A.

Cancel Running Report

The **/api/2.0/fo/report** resource with the parameter **action=cancel** is used to cancel a running report in the user's Report Share. This f unction may be used to cancel a running scorecard report. Authentication is required for each API request. See Chapter 2, "Authentication Using the V2 APIs."

The POST access method must be used to make a request to cancel a running report.

User permissions for canceling running reports are described below.

User Role	Permissions
Manager	Cancel all running reports in subscription.
Unit Manager	Cancel running reports in user's business unit, including reports launched by the user and reports launched by other users in the same business unit.
Scanner	Cancel running reports launched by the user.
Reader	Cancel running reports launched by the user.

Parameters

The parameters used to cancel a running report are described below.

Parameter	Description
action=cancel	(Required) A flag used to make a request to cancel a running report.
id={value}	(Required) Specifies the report ID of a running report that you want to cancel. The status of the report must be "running".
echo_request={**0**|1}	(Optional) Specifies whether to echo the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.

Example

A sample API call to cancel a running report in Report Share (POST method) is shown below.

```
curl -H "X-Requested-With: Curl Sample"
-d "action=cancel&id=1462"
-b "QualysSession=71e6cda2a35d2cd404cddaf305ea0208; path=/api;
secure" "https://qualysapi.qualys.com/api/2.0/fo/scan/"
```

XML Output

A cancel report request (parameter **`action=cancel`**) returns XML output using the DTD "simple_return.dtd", which can be found at the following URL:

```
https://qualysapi.qualys.com/api/2.0/simple_return.dtd
```

The DTD for the simple return XML output is provided in Appendix A.

Download Saved Report

The **/api/2.0/fo/report** resource with **action=fetch** is used to download a saved report in the user's account. You can download all report types (map, scan, patch, authentication, scorecard, remediation, compliance). Just tell us the ID of the report you want to download. The GET or POST access method may be used to make a request.

Authentication is required for each API request. See Chapter 2, "Authentication Using the V2 APIs."

User permissions are below.

User Role	Permissions
Manager	Download all saved reports in subscription.
Unit Manager	Download saved reports in user's business unit, including reports launched by the user and reports launched by other users in the same business unit.
Scanner	Download saved reports launched by the user.
Reader	Download saved reports launched by the user.

Parameters

Use these parameters:

Parameter	Description
action=fetch	(Required) A flag used to make a request to download (fetch) a saved report.
id={value}	(Required) Specifies the report ID of a saved report that you want to download. The status of the report must be "finished".
echo_request={**0**|1}	(Optional) Specify 1 to view input parameters in the XML output. When not specified, parameters are not included in the XML output.

Where do I get the report ID?

Run the report list API

API request:

```
curl -X POST -H X-Requested-With:POSTMAN -H Authorization:Basic
cXV---= -F action=list
https://qualysapi.qualys.com/api/2.0/fo/report/
```

Response:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE REPORT_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/report/report_list_output
.dtd">
<REPORT_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2014-07-02T15:29:52Z</DATETIME>
    <REPORT_LIST>
      <REPORT>
        <ID>7592049</ID>
        <TITLE><![CDATA[FIXED Vuln Report]]></TITLE>
        <TYPE>Scan</TYPE>
        <USER_LOGIN>acme_ur15</USER_LOGIN>
        <LAUNCH_DATETIME>2014-07-02T14:52:45Z</LAUNCH_DATETIME>
        <OUTPUT_FORMAT>HTML</OUTPUT_FORMAT>
        <SIZE>-</SIZE>
        <STATUS>
          <STATE>Running</STATE>
          <MESSAGE><![CDATA[Rendering...]]></MESSAGE>
          <PERCENT>80</PERCENT>
        </STATUS>
<EXPIRATION_DATETIME>2014-07-30T14:52:48Z</EXPIRATION_DATETIME>
      </REPORT>
...
      <REPORT>
        <ID>7589800</ID>
        <TITLE><![CDATA[My Authentication Report]]></TITLE>
        <TYPE>Authentication</TYPE>
        <USER_LOGIN>acme_ee17</USER_LOGIN>
        <LAUNCH_DATETIME>2014-07-02T07:00:21Z</LAUNCH_DATETIME>
        <OUTPUT_FORMAT>PDF</OUTPUT_FORMAT>
        <SIZE>15 KB</SIZE>
        <STATUS>
          <STATE>Finished</STATE>
        </STATUS>
        <EXPIRATION_DATETIME>2014-07-
30T07:00:24Z</EXPIRATION_DATETIME>
      </REPORT>
    </REPORT_LIST>
  </RESPONSE>
```

```
</REPORT_LIST_OUTPUT>
```

Another option - go to the user interface

Within the user interface find the report you want to download (go to Reports > Reports) then choose View Report. In the Report Information window, at the top you'll see the ID in the window URL after id= like this:

`https://qualysguard.qualys.qualys.com/fo/report/view_report.php?id=`**`2281953`**

API request

Sample API request:

```
curl -H "X-Requested-With: Curl Sample"
-b "QualysSession=71e6cda2a35d2cd404cddaf305ea0208; path=/api;
secure" "https://qualysapi.qualys.com/api/2.0/fo/report/
?action=fetch&id=1462"
```

XML output

The XML output uses a DTD for the report you've downloading. For example, if you download a scan report with host based findings (automatic data) then the report uses this DTD (where qualysapi.qualys.com is your Qualys API platform):

`https://qualysapi.qualys.com/asset_data_report.dtd`

Learn more Check out descriptions of various report DTDs in our user guides.

Report	DTD (description)
Scan Results	scan-1.dtd (API v1 User Guide, Appendix A)
Scan Report Host with Based Findings (automatic)	asset_data_report.dtd (API v1 User Guide, Appendix D)
Map Results	map.dtd (API v1 User Guide, Appendix B)
Scorecard Report	multiple DTDs (Appendix E)
Authentication Report	compliance_authentication_report.dtd (Appendix C)
Compliance Policy Report	compliance_policy_report.dtd (Appendix C)

Delete Saved Report

The **/api/2.0/fo/report** resource with the parameter **action=delete** is used to delete a saved report in the user's Report Share. This function may be used to delete a saved scorecard report in the user's Report Share. Authentication is required for each API request. See "Authentication Using the V2 APIs."

The POST access method must be used to make a request to delete a saved report.

User permissions for deleting saved reports are described below.

User Role	Permissions
Manager	Delete all saved reports in the subscription.
Unit Manager	Delete saved reports in user's business unit, including reports launched by the user and reports launched by other users in the same business unit.
Scanner	Delete saved reports launched by the user.
Reader	Delete saved reports launched by the user.

Parameters

The parameters used to delete a saved report are described below.

Parameter	Description
action=delete	(Required) A flag used to make a request to delete a saved report.
id={value}	(Required) Specifies the report ID of a saved report in Report Share that you want to delete. The status of the report must be "finished".
echo_request={**0** | 1}	(Optional) Specifies whether to echo the request's input parameters in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.

Example

A sample API call to download a saved report in Report Share (POST method) is shown below.

```
curl -H "X-Requested-With: Curl Sample"
-d "action=delete&id=1234"
-b "QualysSession=71e6cda2a35d2cd404cddaf305ea0208; path=/api;
secure" "https://qualysapi.qualys.com/api/2.0/fo/scan/"
```

XML Output

A delete report request (parameter **`action=delete`**) returns XML output using the DTD "simple_return.dtd", which can be found at the following URL:

```
https://qualysapi.qualys.com/api/2.0/simple_return.dtd
```

The DTD for the simple return XML output is provided in Appendix A.

CHAPTER 5

Asset API

The Asset API provides information about the hosts in the user account. The host list API function is used to list information about scanned hosts in the user account, including hosts assigned to the VM application and/or PC application.

This chapter describes how to use the Asset API, which is built on the API V2 Architecture. These APIs are covered:

- IP List
- Host List
- Host List Detection
- Excluded Hosts List
- Excluded Hosts Change History
- Manage Excluded Hosts
- Purge Hosts
- Virtual Host List
- Take Actions on Virtual Hosts
- Restricted IPs List
- Add IPs
- Update IPs
- Asset Group API

IP List

The IP List API v2 (**`/api/2.0/fo/asset/ip/?action=list`**) is used to view a list of IP addresses in the user account. By default, all hosts in the user account are included. Optional input parameters support filtering the list by IP addresses and host tracking method. The GET or POST access method may be used to make an API request.

Express Lite: This API is available to Express Lite users.

Authentication is required for each API request. See Chapter 2, "Authentication Using the V2 APIs."

Permissions

User permissions are described below.

User Role	Permissions
Manager	View IP addresses for all hosts in subscription.
Auditor	View IP addresses for all compliance hosts in subscription.
Unit Manager	View IP addresses in user's business unit. To view IP addresses for compliance hosts, the "Manage compliance" permission must be granted in the user's account.
Scanner	View IP addresses in user's account. To view IP addresses for compliance hosts, the "Manage compliance" permission must be granted in the user's account.
Reader	View IP addresses in user's account. To view IP addresses for compliance hosts, the "Manage compliance" permission must be granted in the user's account.

Parameters

The parameters used to make an IP list request are described below.

Parameter	Description
action=list	(Required) A flag used to make an IP list request.
echo_request={**0**|1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When unspecified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
ips={value}	(Optional) Show only certain IP addresses/ranges. One or more IPs/ranges may be specified. Multiple entries are comma separated. A host IP range is specified with a hyphen (for example, 10.10.10.44-10.10.10.90).

Parameter	Description
network_id={value}	(Optional, and valid only when the Network Support feature is enabled for the user's account) Restrict the request to a certain custom network ID.
tracking_method={value}	(Optional) Show only IP addresses/ranges which have a certain tracking method. A valid value is: IP, DNS, or NETBIOS.
compliance_enabled={0|1}	(Optional) Specifying this parameter is valid only when the policy compliance module is enabled for the user account. This parameter is invalid for an Express Lite user. Use this parameter to filter the IP list to show either: 1) a list of compliance IP addresses, or 2) a list of vulnerability management IP addresses. Specify 1 to list compliance IP addresses in the user's account. These hosts are assigned to the policy compliance module. Specify 0 to list hosts which are not assigned to the policy compliance module. An error is returned if a user specifies this parameter, and the user's account does not have compliance management privileges to view the requested list. This may be due to the user's role and/or account settings as indicated below. For a Unit Manager, Scanner or Reader, the "Manage compliance" permission must be enabled in the user account. If the user does not have this permission and sets this parameter to 1, an error is returned. An Auditor user cannot make a request to view vulnerability management IP addresses. If an Auditor sets this parameter to 0, an error is returned.

XML Output

DTD for IP List

An API request to view an IP list returns XML output using the DTD which can be found at the following URL:

```
https://<qualysapi.qualys.com>/api/2.0/fo/
asset/ip/ip_list_output.dtd
```

where <qualysapi.qualys.com> identifies the API server URL where your account is located.

The DTD for the IP list XML is provided in Appendix B.

Sample IP List

Sample IP list output is below.

```
<!DOCTYPE IP_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/asset/ip/
ip_list_output.dtd">

<IP_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2008-05-21T13:32:17Z</DATETIME>
    <IP_SET>
      <IP>123.123.45.0</IP>
      <IP_RANGE>123.124.45.0-123.124.45.255</IP_RANGE>
      <IP_RANGE>123.124.46.0-123.124.46.255</IP_RANGE>
      <IP_RANGE>123.124.47.0-123.124.47.255</IP_RANGE>
      <IP_RANGE>123.124.48.0-123.124.48.255</IP_RANGE>
    </IP_SET>
  </RESPONSE>
</IP_LIST_OUTPUT>
```

Host List

The Host List API v2 (resource **`/api/2.0/fo/asset/host/`** with parameter **`action=list`**) is used to view a list of scanned hosts in the user account. By default, all scanned hosts in the user account are included and basic information about each host is provided. Hosts in the XML output are sorted by host ID in ascending order. Optional input parameters support filtering the list. The GET or POST access method may be used to make an API request.

The output of the Host List API is paginated. By default, a maximum of 1,000 host records are returned per request. You can customize the page size (i.e. the number of host records) by using the parameter "truncation_limit=10000" for instance. In this case the results will be return with pages of 10,000 host records.

Express Lite: This API is available to Express Lite users.

Authentication is required for each API request. See Chapter 2, "Authentication Using the V2 APIs."

User Permissions

User permissions are described below.

User Role	Permissions
Manager	View all scanned hosts in subscription.
Auditor	View all scanned compliance hosts in subscription.
Unit Manager	View scanned hosts in user's business unit. To view compliance hosts, the "Manage compliance" permission must be granted in the user's account.
Scanner	View scanned hosts in user's account. To view compliance hosts, the "Manage compliance" permission must be granted in the user's account.
Reader	View scanned hosts in user's account. To view compliance hosts, the "Manage compliance" permission must be granted in the user's account.

Parameters

The parameters used to make a host list request are described below.

Parameter	Description
action=list	(Required) A flag used to make a host list request.
echo_request={**0**|1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When unspecified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
truncation_limit={value}	(Optional) Specify the maximum number of host records processed per request. When not specified, the truncation limit is set to 1000 host records. You may specify a value less than the default (1-999) or greater than the default (1001-1000000). If the requested list identifies more host records than the truncation limit, then the XML output includes the <WARNING> element and the URL for making another request for the next batch of host records. See ""Sample Record Limit Exceeded Warning". You can specify truncation_limit=0 for no truncation limit. This means that the output is not paginated and all the records are returned in a single output. WARNING: This can generate very large output and processing large XML files can consume a lot of resources on the client side. In this case it is recommended to use the pagination logic and parallel processing. The previous page can be processed while the next page is being downloaded.

Parameter	Description
details={**Basic**|Basic/AGs | All|All/AGs | None}	(Optional) Show the requested amount of host information for each host. A valid value is: Basic, Basic/AGs, All, All/AGs, or None.
	Basic (default) Show basic host information. Basic host information includes the host ID, IP address, tracking method, DNS and NetBIOS hostnames, and operating system.
	Basic/AGs Show basic host information plus asset group information. Asset group information includes the asset group ID and title.
	All. Show all host information. All host information includes the basic host information plus the last vulnerability and compliance scan dates.
	All/AGs. Show all host information plus asset group information. Asset group information includes the asset group ID and title.
	None. Show only the host ID.
ips={value}	(Optional) Show only certain IP addresses/ranges. One or more IPs/ranges may be specified. Multiple entries are comma separated. An IP range is specified with a hyphen (for example, 10.10.10.1-10.10.10.100).
ids={value}	(Optional) Show only certain host IDs/ranges. One or more host IDs/ranges may be specified. Multiple entries are comma separated. A host ID range is specified with a hyphen (for example, 190-400).Valid host IDs are required.
ag_ids={value}	(Optional) Show only hosts belonging to asset groups with certain IDs. One or more asset group IDs and/or ranges may be specified. Multiple entries are comma separated. A range is specified with a dash (for example, 386941-386945). Valid asset group IDs are required.
ag_titles={value}	(Optional) Show only hosts belonging to asset groups with certain strings in the asset group title. One or more asset group titles may be specified. Multiple entries are comma separated (for example, My+First+Asset+Group,Another+Asset+Group).
id_min={value}	(Optional) Show only hosts which have a minimum host ID value. A valid host ID is required.
id_max={value}	(Optional) Show only hosts which have a maximum host ID value. A valid host ID is required.
network_ids={value}	(Optional, and valid only when the Network Support feature is enabled for the user's account) Restrict the request to certain custom network IDs. Multiple network IDs are comma separated.

Parameter	Description
no_vm_scan_since={date}	(Optional) Show hosts not scanned since a certain date and time (optional). The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT), like “2007-07-01” or “2007-01-25T23:12:00Z”. Permissions: An Auditor cannot specify this parameter.
no_compliance_scan_since={date}	(Optional) Show compliance hosts not scanned since a certain date and time (optional). This parameter is invalid for an Express Lite user. The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT), like “2007-07-01” or “2007-01-25T23:12:00Z”. Permissions: A sub-account (Unit Manager, Scanner or Reader) can specify this parameter only when the user is granted permissions to manage compliance information.
vm_scan_since={date}	(Optional) Show hosts that were last scanned for vulnerabilities since a certain date and time (optional). Hosts that were the target of a vulnerability scan since the date/time will be shown. Date/time is specified in this format: YYYY-MM-DD[THH:MM:SSZ] (UTC/GMT). Permissions: An Auditor cannot specify this parameter.
compliance_scan_since={date}	(Optional) Show hosts that were last scanned for compliance since a certain date and time (optional). Hosts that were the target of a compliance scan since the date/time will be shown. This parameter is invalid for an Express Lite user. Date/time is specified in this format: YYYY-MM-DD[THH:MM:SSZ] (UTC/GMT). Permissions: A sub-account (Unit Manager, Scanner or Reader) can specify this parameter only when the user is granted permissions to manage compliance information.

Parameter	Description
compliance_enabled={0|1}	(Optional) This parameter is valid only when the policy compliance module is enabled for the user account. This parameter is invalid for an Express Lite user. Use this parameter to filter the scanned hosts list to show either: 1) a list of scanned compliance hosts, or 2) a list of scanned vulnerability management hosts. Specify 1 to list scanned compliance hosts in the user's account. These hosts are assigned to the policy compliance module. Specify 0 to list scanned hosts which are not assigned to the policy compliance module. A user can specify 0 only when the user has compliance management privileges. For a Unit Manager, Scanner or Reader, the "Manage compliance" permission must be enabled in the user account. If this permission is not enabled and the user makes a request with this parameter set to 0, the request fails with an error (unknown parameter).
os_pattern={expression}	(Optional) Show only hosts which have an operating system matching a certain regular expression. An empty value cannot be specified. Use "%5E%24" to match empty string. **Important Note:** The regular expression string you enter must follow the PCRE standard and it must be URL encoded. See page 149 for sample regular expression strings for matching operating system names. For information about the Perl Compatible Regular Expressions (PCRE) standard visit: http://php.net/manual/en/book.pcre.php For the PCRE syntax, see: http://php.net/manual/en/reference.pcre.pattern.syntax.php http://www.php.net/manual/en/reference.pcre.pattern.posix.php

Report on hosts using asset tags

The input parameters below allow you to report on hosts using asset tags.

Parameter	Description
use_tags={**0** | 1}	(Optional) Specify 0 (the default) if you want to select hosts based on IP addresses/ranges and/or asset groups. Specify **1** if you want to select hosts based on asset tags.
tag_set_by={**id** | name}	(Optional when use_tags=1) Specify "id" (the default) to select a tag set by providing tag IDs. Specify "name" to select a tag set by providing tag names.
tag_include_selector={**any** | all}	(Optional when use_tags=1) Select "any" (the default) to include hosts that match at least one of the selected tags. Select "all" to include hosts that match all of the selected tags.
tag_exclude_selector={**any** | all}	(Optional when use_tags=1) Select "any" (the default) to exclude hosts that match at least one of the selected tags. Select "all" to exclude hosts that match all of the selected tags.
tag_set_include={value}	(Optional when use_tags=1) Specify a tag set to include. Hosts that match these tags will be included. You identify the tag set by providing tag name or IDs. Multiple entries are comma separated.
tag_set_exclude={value}	(Optional when use_tags=1) Specify a tag set to exclude. Hosts that match these tags will be excluded. You identify the tag set by providing tag name or IDs. Multiple entries are comma separated.
show_tags={**0** | 1}	(Optional) Specify 1 to display asset tags associated with each host in the XML output.

XML Output

DTD for Host List

An API request to view a host list returns XML output using the DTD which can be found at the following URL:

```
https://qualysapi.qualys.com/api/2.0/fo/asset/host/host_list_output.dtd
```

where <qualysapi.qualys.com> identifies the API server URL where your account is located.

The DTD for the host list XML is provided in Appendix B.

Sample Record Limit Exceeded Warning

A truncated response is returned when the API request returns more host records than the truncation limit. In this case 1,000 host records are included in the XML output and the Warning message (shown below) indicates the URL you need to use to request the next 1,000 host records.

```
<RESPONSE>
...
   <WARNING>
      <CODE>1980</CODE>
      <TEXT>1000 record limit exceeded. Use URL to get next
batch of results.</TEXT>

<URL><![CDATA[https://qualysapi.qualys.com/api/2.0/fo/asset/hos
t/?action=list&id_min=2400356]]></URL>
   </WARNING>
</RESPONSE>
...
```

Host List Detection

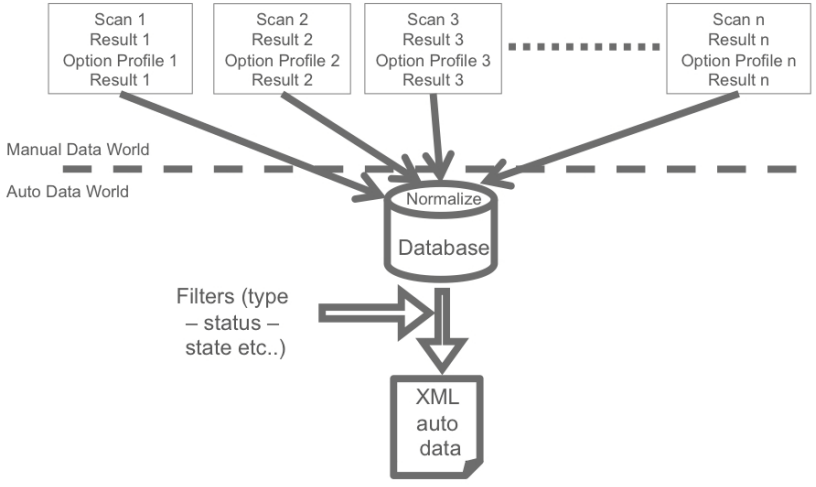
The Host List Detection API v2 (resource **`/api/2.0/fo/asset/host/vm/detection/`** with parameter **`action=list`**) gives API users the ability to obtain "automatic" vulnerability detection data that can be easily imported into a third party solution. Using the host detection API users can request a list of hosts with the hosts latest vulnerability data, based on the "automatic" data available in the user's account. The GET or POST access method may be used to make an API request.

Authentication is required for each API request. See Chapter 2, "Authentication Using the V2 APIs."

Express Lite: This API is available to Express Lite users.

The new host list detection API is recommended as a replacement for other Qualys APIs when the API user wants to manage "automatic" data and integrate this with third party applications. For many third party applications this API is a replacement for the following existing Qualys APIs: **`asset_range_info.php`**, **`asset_data_report.php, asset_search.php`** and **`get_host_info.php`**.

Qualys normalizes the vulnerability scan results into the database using a complex and sophisticated process. This mechanism generates what is called the vulnerability "automatic data". Automatic data brings a lot of value to customers because they provide the latest complete vulnerability status for the hosts (NEW, ACTIVE, FIXED, REOPENED) and history information. Automatic data is completely independent of scan results and option profiles, as shown in the diagram below.



The Qualys database stores automatic data for VM scanned hosts. For each of these hosts there can be multiple detection records.

What is a VM Scanned Host? A VM scanned host is a host that has been successfully scanned by the Qualys VM service for vulnerabilities. Note that a host is considered successfully scanned when it was included as a scan target, the scan was launched and it completed successfully.

What is a Detection Record? A detection record is a unique instance of a discovered vulnerability for a given host. It identifies the host IP address, QID, port, service, FQDN and SSL flag (whether the vulnerability was detected over SSL).

Use Cases

The detection API is most likely used in conjunction with other information that can be downloaded using other Qualys APIs.

Create Custom Technical Reports with vulnerability details

Technical reports need additional information for each vulnerability such as the description, solution, threat or impact. The detection API provides the QID for each vulnerability found for an asset. The QID is a unique ID that references a vulnerability within the Qualys KnowledgeBase.

Use the following workflow to create custom technical reports:

Step 1. Use the host list detection API to return "automatic" vulnerability data for hosts in your account, as described in this section.

Step 2. Use the KnowlegeBase API (/api/2.0/fo/knowledge_base/vuln/?action=list) to obtain vulnerability data, such as the vulnerability description, threat and impact. It's possible to make a request for all vulnerabilities (QIDs) in the KnowledgeBase or just a specific vulnerability.

For example, to make a request for QID 90082 use the following URL:

```
https://qualysapi.qualys.com/api/2.0/fo/knowledge_base/vuln/?action=list&
ids=90082
```

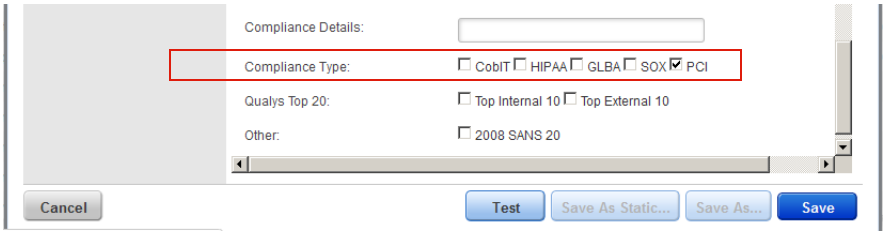
where "qualysapi.qualys.com" is the name of the API server where your account is located (in this case US Platform 1). If your account is located on another platform please insert the appropriate platform URL. For example "qualysapi.qualys.eu" if your account is located on the EU platform.

Step 3. Correlate the vulnerability information in the third party application using the QID number provided in the <QID> XML output which is returned by the host detection API (Step 1) and the KnowledgeBase API (Step 2).

A typical integration would be to create tables in a database for the XML output from both Qualys API functions and use QID as a key for a join. This way it would be possible to create queries that will provide all the vulnerabilities for a given set of hosts (according to custom search criteria) and their descriptions.

Get All PCI Vulnerabilities

Step 1. First you need to create a dynamic search list titled "PCI Vulns" using the Qualys user interface. When creating the dynamic search list, select the PCI option next to Compliance Type as shown below.



Step 2. Create an asset group titled "PCI Hosts" containing the hosts which are in scope for PCI compliance.

Step 3. Make the following host list detection API request using the asset group title "PCI Hosts" and the search list title "PCI Vulns":

```
https://qualysapi.qualys.com/api/2.0/fo/asset/host/vm/detection/?action=list&ag_titles=PCI+Hosts&include_search_list_titles=PCI+Vulns'
```

where "qualysapi.qualys.com" is the name of the API server where your account is located (in this case US Platform 1). If your account is located on another platform please insert the appropriate platform URL. For example "qualysapi.qualys.eu" if your account is located on the EU platform.

Asset Search Portal Capability

The new host detection API endpoint allows users to do the same kind of searches as the Asset Search Portal user interface (under Assets > Asset Search) using these search parameters:

- Asset Groups
- IPs/Ranges
- Operating System
- QID

The host detection API offers users more search parameters and capability than the Asset Search user interface. Using the API users can get the details of vulnerability detections. Also, the API allows users to do more advanced searches on QIDs using all the power of search lists.

User Permissions

User permissions are described below.

User Role	Permissions
Manager	View all VM scanned hosts in subscription.
Auditor	No permissions to view VM scanned hosts.
Unit Manager	View VM scanned hosts in user's business unit.
Scanner	View VM scanned hosts in user's account.
Reader	View VM scanned hosts in user's account.

Input Parameters

The input parameter **`action=list`** is required. All other input parameters are optional. Several filtering parameters are provided for filtering hosts and QIDs. When multiple filter parameters are specified, the service combines the effects of all the parameters in a way that corresponds to a logical "AND". So if two filter parameters are specified in the request, the service returns hosts that match both filters.

API Request. These input parameters are used for making the API request.

Parameter	Description
action=list	(Required) Specify **`action=list`** to view a list of detection records for VM scanned hosts.
echo_request={0|1}	(Optional) Show (echo) the request's input parameters (names and values) in the output. When unspecified, parameters are not included in the output. Specify 1 to view parameters in the output.
active_kernels_only={**0**|1}	(Optional) Specify 1 to exclude vulnerabilities found on non-running Linux kernels from the XML output. This option is available when the New Data Security Model is enabled in the user's subscription. How it works: It's possible that multiple kernels are detected on a single Linux host. You'll notice the scan results report the running kernel on each Linux host in Information Gathered QID 45097. When you make a "Detection" API request with "active_kernels_only=1" the XML output includes only those vulnerabilities found on the running kernel.

Parameter	Description
output_format={**XML** | CSV | CSV_NO_METADATA}	(Optional) Specifies the format of the host detection list output. When not specified, the output format is XML. A valid value is XML, CSV, or CSV_NO_METADATA. XML (default). Specifies XML format for the output. CSV. Specifies CSV format for the output. The output is structured in these sections: HEADER_CSV (lists input parameters specified during the list request if **`echo_request=1`** is also specified), BODY_CSV (lists host records matching filters) and FOOTER_CSV (lists status messages and truncation details, if applicable). CSV_NO_METADATA. Specifies CSV format for the output with no metadata. In this case, the output will not be structured with header, body and footer sections, and will not indicate whether the list is truncated.
suppress_duplicated_data_from_csv={**`0`** | 1}	(Optional) By default or when set to 0, host details will be repeated in each line of detection information in the XML output. When set to 1, host details will not be repeated (suppressed) in each detection line. This parameter must be specified with: **`output_format=CSV`** or **`output_format=CSV_NO_METADATA`**.
truncation_limit={value}	(Optional) Specifies the maximum number of host records processed per request. When not specified, the truncation limit is set to 1000 host records. You may specify a value less than the default (1-999) or greater than the default (1001-1000000). Specify 0 for no truncation limit. If the requested list identifies more host records than the truncation limit and **`output_format=XML`**, then the XML output includes the <WARNING> element and the URL for making another request for the next batch of host records. If the requested list identifies more host records than the truncation limit and **`output_format=CSV`**, then the CSV output includes "Truncated" in the FOOTER_CSV section and the URL for making another request for the next batch of host records. See these sections for further information: "Sample Host List Detection XML Output — Truncation" "Sample Host List Detection CSV Output — Truncation" "Sample Script for Pagination Logic"

Host Filters. These input parameters are used to filter hosts. All of these parameters are optional.

Parameter	Description
ids={value}	(Optional) Show only certain host IDs/ranges. One or more host IDs/ranges may be specified. Multiple entries are comma separated. A host ID range is specified with a hyphen (for example: 190-400).Valid host IDs are required.
id_min={value}	(Optional) Show only hosts which have a minimum host ID value.
id_max={value}	(Optional) Show only hosts which have a maximum host ID value. A valid host ID is required.
ips={value}	(Optional) Show only certain IP addresses/ranges. One or more IPs/ranges may be specified. Multiple entries are comma separated. An IP range is specified with a hyphen (for example: 10.10.10.1-10.10.10.100).
ag_ids={value}	(Optional) Show only hosts belonging to asset groups with certain IDs. One or more asset group IDs and/or ranges may be specified. Multiple entries are comma separated. A range is specified with a dash (for example: 386941-386945). Valid asset group IDs are required.
	The **ag_ids** and **ag_titles** parameters are mutually exclusive and cannot be specified together in the same request.
ag_titles={value}	(Optional) Show only hosts belonging to asset groups with certain strings in the asset group title. One or more asset group titles may be specified. Multiple entries are comma separated (for example, My+First+Asset+Group,Another+Asset+Group).
	The **ag_ids** and **ag_titles** parameters are mutually exclusive and cannot be specified together in the same request.
network_ids={value}	(Optional, and valid only when the Network Support feature is enabled for the user's account) Restrict the request to certain custom network IDs. Multiple network IDs are comma separated.
vm_scan_since={date}	(Optional) Show hosts scanned and processed since a certain date and time (optional). The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT), like "2007-07-01" or "2007-01-25T23:12:00Z".
	This parameter cannot be specified with **max_days_since_vm_scan** in the same request.
no_vm_scan_since={date}	(Optional) Show hosts not scanned and processed since a certain date and time (optional). The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT), like "2007-07-01" or "2007-01-25T23:12:00Z".

Parameter	Description
	This parameter cannot be specified with **max_days_since_vm_scan** in the same request.
max_days_since_last_vm_scan={value}	(Optional) Show only hosts scanned and processed in the past number of days, where the value is a number of days. This parameter cannot be specified with any of these parameters in the same request: **vm_scan_since** and **no_vm_scan_since**.
status={value}	(Optional) Show only hosts with one or more of these status values: New, Active, Re-Opened, Fixed. Multiple status values are entered as a comma-separated list.
compliance_enabled={0|1}	(Optional) This parameter is valid only when the policy compliance module is enabled for the user account. This parameter is invalid for an Express Lite user. Specify 1 to list compliance hosts in the user's account that have been scanned and processed. These hosts are assigned to the policy compliance module. Specify 0 to list scanned hosts which are not assigned to the policy compliance module.
os_pattern={expression}	(Optional) Show only hosts which have an operating system matching a certain regular expression. An empty value cannot be specified. Use "%5E%24" to match empty string. **Important Note:** The regular expression string you enter must follow the PCRE standard and it must be URL encoded. See page 149 for sample regular expression strings for matching operating system names. For information about the Perl Compatible Regular Expressions (PCRE) standard visit: http://php.net/manual/en/book.pcre.php For the PCRE syntax, see: http://php.net/manual/en/reference.pcre.pattern.syntax.php http://www.php.net/manual/en/reference.pcre.pattern.posix.php

QID Filters. These input parameters are used to filter QIDs. All of these parameters are optional.

Parameter	Description
qids={value}	(Optional) Show only detection records with certain QIDs. One or more QIDs may be specified. A range is specified with a dash (for example: 68518-68522). Multiple entries are comma separated. Valid QIDs are required.
severities={value}	(Optional) Show only detection records which have certain severities. One or more levels may be specified. A range is specified with a dash (for example: 1-3). Multiple entries are comma separated.
show_igs={0|1}	(Optional except as noted) Specify 1 to show detection records with information gathered along with confirmed vulnerabilities and potential vulnerabilities. Specify 0 (default) to hide information gathered.
	The **show_igs** parameter is required in one use case. The parameter **show_igs=1** must be specified if both these conditions are met: 1) search lists are included using the parameter **include_search_list_titles** or **include_search_list_ids**, and 2) if the included search lists contain only information gathered.
include_search_list_titles={value}	(Optional) Show detection records only when a record's QID is INCLUDED IN one or more of the specified search list titles. One or more titles may be specified. Multiple titles are comma separated.
	This parameter cannot be specified with any of these parameters in the same request: **qids**, **severities** or **include_search_list_ids**.
exclude_search_list_titles={value}	(Optional) Show detection records only when a record's QID is IS EXCLUDED from one or more of the specified search list titles. One or more titles may be specified. Multiple titles are comma separated.
	This parameter cannot be specified with any of these parameters in the same request: **qids**, **severities** or **exclude_search_list_ids**.
include_search_list_ids={value,value...}	(Optional) Show detection records only when a record's QID IS INCLUDED in one or more of the specified search list titles. One or more IDs may be specified. A range is specified with a dash (for example: 10-15). Multiple entries are comma separated.
	This parameter cannot be specified with any of these parameters in the same request: **qids**, **severities** or **include_search_list_titles**.

Parameter	Description
exclude_search_list_ids={value,value...}	(Optional) Show detection records only when a record's QID IS EXCLUDED from one or more of the specified search list titles. One or more IDs may be specified. A range is specified with a dash (for example: 40-42). Multiple entries are comma separated.
	This parameter cannot be specified with any of these parameters in the same request: **`qids`**, **`severities`** or **`exclude_search_list_titles`**.

Report on hosts using asset tags. The input parameters below allow you to report on hosts using asset tags.

Parameter	Description
use_tags={**`0`** | 1}	(Optional) Specify 0 (the default) if you want to select hosts based on IP addresses/ranges and/or asset groups. Specify **`1`** if you want to select hosts based on asset tags.
tag_set_by={**`id`** | name}	(Optional when use_tags=1) Specify "id" (the default) to select a tag set by providing tag IDs. Specify "name" to select a tag set by providing tag names.
tag_include_selector={**`any`** | all}	(Optional when use_tags=1) Select "any" (the default) to include hosts that match at least one of the selected tags. Select "all" to include hosts that match all of the selected tags.
tag_exclude_selector={**`any`** | all}	(Optional when use_tags=1) Select "any" (the default) to exclude hosts that match at least one of the selected tags. Select "all" to exclude hosts that match all of the selected tags.
tag_set_include={value}	(Optional when use_tags=1) Specify a tag set to include. Hosts that match these tags will be included. You identify the tag set by providing tag name or IDs. Multiple entries are comma separated.
tag_set_exclude={value}	(Optional when use_tags=1) Specify a tag set to exclude. Hosts that match these tags will be excluded. You identify the tag set by providing tag name or IDs. Multiple entries are comma separated.
show_tags={**`0`** | 1}	(Optional) Specify 1 to display asset tags associated with each host in the XML output.

Keep Alive Mechanism

The service uses a "keep alive" mechanism to maintain an open connection to the Qualys server for the duration of the host detection list API request. To keep the connection alive, the service sends some "dummy" data back to the client every 30 to 40 seconds if no "real" data has been sent already by the API during that time.

In XML output, this "dummy" data appears as a "<!-- keep-alive -->" line (since comments should be safely ignored by downstream XML parsers).

In CSV and CSV_NO_METADATA output, this "dummy" data appears as a <CR><LF> (carriage return, linefeed) pair (since empty lines clearly do not contain any CSV data).

Sample API Requests

These sample requests work on Qualys US Platform 1 where the FQDN in the API server URL is qualysapi.qualys.com. Please be sure to replace the FQDN with the proper API server URL for your platform. For a partner platform, use the URL for your @customer platform API server.

The header parameter "X-Requested-With" is also provided as an example. Please change this parameter according to your need. This parameter is exposed in the activity log (only displayed when used with session cookie authentication - does not apply for basic authentication as used in the following examples) and it is useful for monitoring API activity.

Sample 1. Scanned Hosts in IP Range

Request a list of VM scanned hosts which have IP addresses in the range 10.10.30.0-10.10.30.255:

```
curl -H 'X-Requested-With: Curl Sample' -u 'username:password'
'https://qualysapi.qualys.com/api/2.0/fo/asset/host/vm/detection/?action=
list&ips=10.10.30.0-10.10.30.255'
```

Sample 2. Scanned Hosts with Certain QIDs

Request a list of VM scanned hosts which have QIDs included in the search list title "QID 45038":

```
curl -H 'X-Requested-With: Curl Sample' -u 'username:password'
'https://qualysapi.qualys.com/api/2.0/fo/asset/host/vm/detection/?action=
list&include_search_list_titles=QID+45038'
```

Sample 3. Scanned Hosts in IP Range with Certain QIDs

Request a list of VM scanned hosts which have the IP addresses in the range 10.10.24.100-10.10.24.120 and which have detected QIDs included in "My QID List":

```
curl -H 'X-Requested-With: Curl Sample' -u 'username:password'
'https://qualysapi.qualys.com/api/2.0/fo/asset/host/vm/detection/?action=
list&ips=10.10.24.100-10.10.24.120&
include_search_list_titles=My+QID+List'
```

For the sample request shown above, only hosts that match both filters will be included in the host list output. If a host is included in the IP range 10.10.24.100-10.10.24.120 and the host does not have detected QIDs in "My QID List" then the host will not be listed.

Sample 4. Fixed Detections

Fixed detections can be returned in the output of the detection API. By default, when no specific parameters are passed to this API, the output only contains the detections with New, Active or Re-Opened <STATUS> only. The reason for this is because the primary goal of the detection API is to return the most updated vulnerability status for the hosts. By using the parameter "status=Fixed" like in the example below, it is possible to get the Fixed vulnerabilities:

```
$ curl -u "LOGIN:PASS" -H "X-Requested-With: curl"
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/vm/detection/?action=
list&status=Fixed"

<?xml version="1.0" encoding="UTF-8" ?>
<HOST_LIST_VM_DETECTION_OUTPUT>
  <RESPONSE>
    <DATETIME>2014-06-15T20:05:15Z</DATETIME>
    <HOST_LIST>
      <HOST>
        <ID>2131856</ID>
        <IP>10.10.10.15</IP>
        <TRACKING_METHOD>IP</TRACKING_METHOD>
        <OS><![CDATA[Windows 2003/XP]]></OS>
        <LAST_SCAN_DATETIME>2014-06-11T18:04:43Z</LAST_SCAN_DATETIME>
        <DETECTION_LIST>
          <DETECTION>
            <QID>38094</QID>
            <TYPE>Potential</TYPE>
            <SEVERITY>2</SEVERITY>
            <PORT>3389</PORT>
            <PROTOCOL>tcp</PROTOCOL>
            <SSL>0</SSL>
            <RESULTS><![CDATA[Detected ...]></RESULTS>
            <STATUS>Fixed</STATUS>
```

```
<FIRST_FOUND_DATETIME>2014-06-11T17:40:35Z</FIRST_FOUND_DATETIME>
        <LAST_FOUND_DATETIME>2014-06-11T17:40:35Z</LAST_FOUND_DATETIME>
        <LAST_TEST_DATETIME>2014-06-11T17:40:35Z</LAST_TEST_DATETIME>
        <LAST_FIXED_DATETIME>2014-06-11T18:04:43Z</LAST_FIXED_DATETIME>

          </DETECTION>
          <DETECTION>
            [...]
          </DETECTION>
        </DETECTION_LIST>
      </HOST>
    </HOST_LIST>
  </RESPONSE>
</HOST_LIST_VM_DETECTION_OUTPUT>
```

Please note that this input parameter can also be used to filter out the output for only the desired status. For example:

- status=Fixed only returns the Fixed detections
- status=New,Fixed only returns the New and Fixed detections
- status=New,Fixed,Active,Re-Opened returns All detections

Sample 5. Differential Updates

Differential updates are mainly important for large organizations so they can optimize the size of the XML output and avoid redundant information which could represent the main part of the output. It is important because when the dates are post-processed on the client side, this can consume a lot of resources (CPU, memory mainly) to parse the XML output, and differential data guarantee to only process what changed instead of the redundant information.

Note: In order to leverage the differential update capabilities of the detection API, the user must start the process by doing a full download of all the data and then using the parameter “max_days_since_last_vm_scan” or “vm_scan_since” to only download the data for the hosts that have been scanned and processed since the last call to the API.

The parameter “max_days_since_last_vm_scan=Xdays” can be used to get all the changes that happened during the past X days.

In the following example, the call to the detection API will return all the New, Re-Opened and Active detections for all the hosts that have been scanned during the past 5 days.

```
$ curl -u "LOGIN:PASS" -H "X-Requested-With: curl"
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/vm/detection/?action=
list&max_days_since_last_vm_scan=5"
```

This following request is equivalent, but a little bit more difficult to build because the user must explicitly provides the date of the day 5 days ago. For instance, if the date of today is March, 3rd 2012, the date 5 days ago is February, 27th 2012 (2012 is a leap year).

```
$ curl -u "LOGIN:PASS" -H "X-Requested-With: curl"
“https://qualysapi.qualys.com/api/2.0/fo/asset/host/vm/detection/?action=
list&vm_scan_since=2012-02-27"
```

Unlike the first request that uses the “max_days_since_last_vm_scan” parameter, this request has to be updated every time the call is performed.

Sample 6. CSV Output

By default, the output of the detection API is returned in XML, but when the parameter “output_format=CSV” is provided, then the output is returned into a comma separated value (csv) format. This format is is convenient because it can be opened as a spreadsheet and it is also easier to read for a human.

When the parameter “suppress_duplicated_data_from_csv=0” is provided, then the CSV output is going to be augmented with all the redundant information for each line of the file. By default, when this parameter is not specified and when the CSV format is chosen, the output is trimmed down to remove the redundant information.

```
$ curl -u "LOGIN:PASS" -H "X-Requested-With: curl"
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/vm/detection/?action=
list&output_format=CSV&suppress_duplicated_data_from_csv=0"
```

6	Host ID	IP Address	Tracking Met	Operating Sy	DNS Name	Netbios Nam	Last Scan Da	OS CPE	QID	Type	Port	Protocol	FQDI
7	2131865	10.10.24.127	IP	Linux 2.4-2.6 / Embedded Device / F5 N			2012-04-03T21:20:57Z						
8									11	Confirmed			
9									38019	Confirmed			
10									38560	Potential			
11									42339	Potential			
12									42340	Potential			
13									66040	Potential			
14									82003	Confirmed			
15									82054	Confirmed			

For your information, here is how the CSV output looks like when this option is not used:

6	Host ID	IP Address	Tracking Met	Operating Sy	DNS Name	Netbios Nam	Last Scan Da	OS CPE	QID	Type	Port	Protocol	FQD
7	2131865	10.10.24.127	IP	Linux 2.4-2.6 / Embedded Device / F5 N			2012-04-03T21:20:57Z						
8	Information of the line above is not duplicated here, unlike in								11	Confirmed			
9	the previous using the new parameters								38019	Confirmed			
10	output_format=CSV&suppress_duplicated_data_from_csv=0								38560	Potential			
11									42339	Potential			
12									42340	Potential			
13									66040	Potential			
14									82003	Confirmed			
15									82054	Confirmed			

This CSV format with all the redundant information for each line is also convenient for quick ad-hoc text manipulation with tools such as “grep”, “cut”, “sed”, “awk” etc.

Sample OS Pattern Filters

Important Note: The regular expression string you provide in the "os_pattern" parameter must follow the PCRE standard and it must be URL encoded to make sure that special characters are correctly passed to the filter.

Sample 1. OS Name "Beginning With"

To list hosts in the range 10.10.10.1-10.10.10.100 which have an operating system beginning with "Windows", you can use the following URL:

```
https://qualysapi.qualys.com/api/2.0/fo/asset/host/vm/detection/?action=list&ips=10.10.10.1-10.10.10.100&os_pattern=%5EWindows
```

where "%5EWindows" is the URL encoded version of the PCRE regular expression "^Windows". Any host which was scanned for vulnerabilities and which has an operating system starting with "Windows" will be included in the host list output.

Sample 2. OS Name "Ending With"

To list hosts in the range 10.10.10.1-10.10.10.100 which have the operating system name "FreeBSD", you can use the following URL:

```
https://qualysapi.qualys.com/api/2.0/fo/asset/host/vm/detection/?action=list&ips=10.10.10.1-10.10.10.100&os_pattern=BSD%24
```

where "BSD%24" is the URL encoded version of the PCRE regular expression "BSD$". For the above sample, hosts which are in the range 10.10.10.1-10.10.10.100 and which have an operating system ending in BSD, such as FreeBSD, will be included in the host list output.

Sample 3. OS Name with Simple Pattern

To list hosts which have an operating system name containing "2003" you can use the following URL:

```
https://qualysapi.qualys.com/api/2.0/fo/asset/host/vm/detection/?action=list&os_pattern=2003
```

Any host which was scanned for vulnerabilities and which has an operating system including the string "2003" will be included in the host list output.

Sample 4. OS Name with More Complex Pattern

To list hosts in the range 10.10.10.1-10.10.10.100 which have an operating system name containing a more complex string like "Windows 64 bit" you could use the OS pattern shown in the following URL:

```
https://qualysapi.qualys.com/api/2.0/fo/asset/host/vm/detection/?action=l
ist&ips=10.10.10.1-10.10.10.100&os_pattern=%5EWin.*64%20bit.*Service%20Pa
ck%201%24
```

where "%5EWin.*64%20bit.*Service%20Pack%201%24" is the URL encoded version of the PCRE regular expression "^Win.*64+bit.*Service+Pack+1". The above regular expression matches OS string starting with "Win", containing "64 bit" and ending with "Service Pack 1". Possible matches include hosts in the range 10.10.10.1-10.10.10.100 which have some flavors of Windows 64 Bit Service Pack 1 such as these:

- Windows XP 64 bit Edition Service Pack 1
- Windows 2008 Core 64 bit Edition Service Pack 1
- Windows 2003 Server 64 bit Edition AD Service Pack 1

Sample 5. Hosts with No OS

To list hosts with no operating system defined, you can use the following URL:

```
https://qualysapi.qualys.com/api/2.0/fo/asset/host/vm/detection/?action=l
ist&ips=10.10.10.1-10.10.10.100&os_pattern=%5E%24
```

where "%5E%24" is the URL encoded version of the PCRE regular expression "^$". The above request will list all hosts in the user account that do not have an operating system defined.

XML Output

The DTD for the host list detection output can be found at the following URL:

```
https://qualysapi.qualys.com/api/2.0/fo/asset/host/vm/detection
/host_list_vm_detection_output.dtd
```

The DTD for the XML output is provided in Appendix B.

Sample Host List Detection XML Output

Sample host list detection XML output is shown below.

```
<HOST_LIST_VM_DETECTION_OUTPUT>

  <RESPONSE>
    <DATETIME>2014-06-18T21:41:06Z</DATETIME>
    <HOST_LIST>
      <HOST>
        <ID>2401054</ID>
        <IP>10.10.10.2</IP>
        <TRACKING_METHOD>IP</TRACKING_METHOD>
        <OS><![CDATA[Linux 2.2-2.6]]></OS>
        <DNS><![CDATA[ns2.vuln.qa.qualys.com]]></DNS>
        <LAST_SCAN_DATETIME>2014-06-18T20:12:58Z</LAST_SCAN_DATETIME>
        <DETECTION_LIST>
          <DETECTION>
            <QID>15018</QID>
            <TYPE>Confirmed</TYPE>
            <SEVERITY>3</SEVERITY>
            <PORT>53</PORT>
            <PROTOCOL>tcp</PROTOCOL>
            <SSL>0</SSL>
            <RESULTS><![CDATA[Name    Class   Type    Value
vuln.qa.qualys.com     IN      SOA     bart.vuln.qa.qualys.com
qalab.qualys.com (52 10800 1800 ...]]></RESULTS>
            <STATUS>New</STATUS>
            <FIRST_FOUND_DATETIME>2014-06-
10T10:09:05Z</FIRST_FOUND_DATETIME>
            <LAST_FOUND_DATETIME>2014-06-
18T20:09:44Z</LAST_FOUND_DATETIME>
            <LAST_TEST_DATETIME>2014-06-18T20:09:44Z</LAST_TEST_DATETIME>
            <LAST_UPDATE_DATETIME>2014-06-
18T20:15:45Z</LAST_UPDATE_DATETIME>
          </DETECTION>

          <DETECTION>
            <QID>15019</QID>
            <TYPE>Confirmed</TYPE>
```

```
            <SEVERITY>3</SEVERITY>
            <PORT>53</PORT>
            <PROTOCOL>tcp</PROTOCOL>
            <SSL>0</SSL>
            <RESULTS><![CDATA[10.10.10.1]]></RESULTS>
            <STATUS>New</STATUS>
            <FIRST_FOUND_DATETIME>2014-06-
10T10:00:01Z</FIRST_FOUND_DATETIME>
            <LAST_FOUND_DATETIME>2014-06-
18T20:09:44Z</LAST_FOUND_DATETIME>
            <LAST_TEST_DATETIME>2014-06-18T20:09:44Z</LAST_TEST_DATETIME>
            <LAST_UPDATE_DATETIME>2014-06-
18T20:15:45Z</LAST_UPDATE_DATETIME>
          </DETECTION>

          <DETECTION>
            <QID>15020</QID>
            <TYPE>Confirmed</TYPE>
            <SEVERITY>2</SEVERITY>
            <PORT>53</PORT>
            <PROTOCOL>udp</PROTOCOL>
            <SSL>0</SSL>
            <RESULTS><![CDATA[Name    IP
fw01-corp.us01.qualys.com.     10.0.0.1...]]></RESULTS>
            <STATUS>New</STATUS>
            <FIRST_FOUND_DATETIME>2014-06-
18T20:09:44Z</FIRST_FOUND_DATETIME>
            <LAST_FOUND_DATETIME>2014-06-
18T20:09:44Z</LAST_FOUND_DATETIME>
            <LAST_TEST_DATETIME>2014-06-18T20:09:44Z</LAST_TEST_DATETIME>
            <LAST_UPDATE_DATETIME>2014-06-
18T20:15:44Z</LAST_UPDATE_DATETIME>
          </DETECTION>
        </DETECTION_LIST>
      </HOST>

...
```

Sample Host List Detection XML Output — Truncation

A truncated response is returned when the API request returns more host records than the truncation limit. In this case 1,000 host records are included in the XML output and the Warning message (shown below in **bold**) indicates the URL you need to use to request the next 1,000 host records.

Sample host list detection XML output which has been truncated is shown below.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE HOST_LIST_VM_DETECTION_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/vm/detection/host_lis
t_vm_detection_output.dtd">

<HOST_LIST_VM_DETECTION_OUTPUT>
  <RESPONSE>
    <DATETIME>2014-06-17T17:36:43Z</DATETIME>
    <HOST_LIST>
      <HOST>
        <ID>2400354</ID>
        <IP>10.10.10.12</IP>
        <TRACKING_METHOD>NETBIOS</TRACKING_METHOD>
        <NETBIOS><![CDATA[XPSP2-SERVER]]></NETBIOS>
        <LAST_SCAN_DATETIME>2011-01-17T17:20:55Z</LAST_SCAN_DATETIME>
        <DETECTION_LIST>
          <DETECTION>
            <QID>6</QID>
            <TYPE>Info</TYPE>
            <RESULTS><![CDATA[IP address        Host name
10.10.10.12     No registered hostname]]></RESULTS>
...
          </DETECTION>
      </HOST>

... (998 HOST records excluded)

      <HOST>
...

    </HOST_LIST>
    <WARNING>
      <CODE>1980</CODE>
      <TEXT>1000 record limit exceeded. Use URL to get next batch of
results.</TEXT>
<URL><![CDATA[https://qualysapi.qualys.com/api/2.0/fo/asset/host/vm/dete
ction/?action=list&show_igs=1&id_min=2400356]]></URL>
    </WARNING>
  </RESPONSE>
</HOST_LIST_VM_DETECTION_OUTPUT>
```

Sample Host List Detection CSV Output — Truncation

A truncated response is returned when the API request returns more host records than the truncation limit - in this case this is set to 2. The FOOTER_CSV section (shown below in **bold**) indicates the URL you need to use to request the next 2 host records.

```
----BEGIN_RESPONSE_HEADER_CSV
"Launch Datetime","User Login","Resource","Parameter Name","Parameter
Value"
"2011-03-
28T17:54:16Z","quays_ts20","https://qualysapi.qualys.com/api/2.0/fo/asset
/host/vm/detection/",,
,,,"action","list"
,,,"output_format","CSV"
,,,"echo_request","1"
,,,"truncation_limit","2"
----END_RESPONSE_HEADER_CSV
----BEGIN_RESPONSE_BODY_CSV
"Host ID","IP Address","Tracking Method","Operating System","DNS
Name","Netbios Name","Last Scan Datetime","QID","Type","Severity",
"Port","Protocol","FQDN","SSL","Status","First Found Datetime","Last
Found Datetime","Last Test Datetime","Last Update Datetime","Results"
"2401054","10.10.10.2","IP","Linux 2.2-2.6","ns2.qualys.com",,"2011-01-
18T20:12:58Z",,,,,,,,,,,,
,,,,,,,"15019","Confirmed","3","53","tcp",,"0","New","2011-01-
18T20:09:44Z","2011-01-18T20:09:44Z","2011-01-18T20:09:44Z","2011-01-
18T20:15:45Z","10.10.10.1"
,,,,,,,"15034","Confirmed","2","53","udp",,"0","New","2011-01-
18T20:09:44Z","2011-01-18T20:09:44Z","2011-01-18T20:09:44Z","2011-01-
18T20:15:45Z","Server supports recursive name resolution to IPv4
addresses.
Server supports recursive name resolution to IPv6 addresses."
,,,,,,,"15035","Potential","2","53","udp",,"0","New","2011-01-
18T20:09:44Z","2011-01-18T20:09:44Z","2011-01-18T20:09:44Z","2011-01-
18T20:15:45Z","Server's cache timeout for IPv4 addresses is more than 3
sec.
Server's cache timeout for IPv6 addresses is more than 3 sec."
,,,,,,,"82003","Confirmed","1",,,,"0","New","2011-01-18T20:09:44Z","2011-
01-18T20:09:44Z","2011-01-18T20:09:44Z","2011-01-18T20:15:39Z","Timestamp
of host (network byte ordering): 14:10:57 GMT"
,,,,,,,"82054","Confirmed","2",,,,"0","New","2011-01-18T20:09:44Z","2011-
01-18T20:09:44Z","2011-01-18T20:09:44Z","2011-01-18T20:15:44Z","Tested on
port 22 with an injected SYN/RST offset by 16 bytes.
Tested on port 53 with an injected SYN/RST offset by 16 bytes."
"2401055","10.10.10.15","IP","Windows XP","xpsp2-30","XPSP2-30","2011-01-
18T20:12:59Z",,,,,,,,,,,,
,,,,,,,"38094","Potential","2","3389","tcp",,"0","New","2011-01-
18T20:09:44Z","2011-01-18T20:09:44Z","2011-01-18T20:09:44Z","2011-01-
18T20:15:45Z","Detected service win_remote_desktop and os WINDOWS XP
```

```
SERVICE PACK 2-3"
,,,,,,,"70000","Confirmed","2",,,,"0","New","2011-01-18T20:09:44Z","2011-
01-18T20:09:44Z","2011-01-18T20:09:44Z","2011-01-18T20:15:45Z","XPSP2-30"
,,,,,,,"82003","Confirmed","1",,,,"0","New","2011-01-18T20:09:44Z","2011-
01-18T20:09:44Z","2011-01-18T20:09:44Z","2011-01-18T20:15:45Z","Timestamp
of host (host byte ordering): 20:09:42 GMT"
,,,,,,,"82058","Confirmed","2",,,,"0","New","2011-01-18T20:09:44Z","2011-
01-18T20:09:44Z","2011-01-18T20:09:44Z","2011-01-18T20:15:45Z","Tested on
port 135 with ICMP Destination Unreachable Type 3, Codes 2, 3, &amp;amp; 4
Hard Errors (with a TCP Sequence Offset of 16 Bytes).
Tested on port 445 with ICMP Destination Unreachable Type 3, Codes 2, 3,
&amp;amp; 4 Hard Errors (with a TCP Sequence Offset of 16 Bytes)."
----END_RESPONSE_BODY_CSV
----BEGIN_RESPONSE_FOOTER_CSV
"Status Message","Next ID","Next URL"
"Truncated","2401056","https://qualysapi.qualys.com/api/2.0/fo/asset/host
/vm/detection/?action=list&output_format=CSV&echo_request=1&truncation_li
mit=2&id_min=2401056"
----END_RESPONSE_FOOTER_CSV
```

Sample Script for Pagination Logic

The following script illustrates how it's possible for a third party application to retrieve paginated results using the host list detection API.

```
#!/bin/sh

BASE_URL='https://qualysapi.qualys.com'
OUTPUT_FILENAME='detection'
USERNAME='username'
PASS='password'
N='01'

# Specify filtering parameters here (Remember to URL encode os_pattern, if
specified):

#PARAMETER="&show_igs=1&include_search_list_titles=QID+45038&os_pattern=%
5Cd%2B"

echo
"${BASE_URL}/api/2.0/fo/asset/host/vm/detection/?action=list${PARAMETER}"
curl -H 'X-Requested-With: curl example QWEB 6.17' -k -u
"${USERNAME}:${PASS}"
"${BASE_URL}/api/2.0/fo/asset/host/vm/detection/?action=list${PARAMETER}"
> "${OUTPUT_FILENAME}-${N}.xml"

while grep '<URL><!\[CDATA\[https:\/\/.*qualys\.com' "${OUTPUT_FILENAME}-
${N}.xml"
```

```
do
  NEXT_URL=`grep "<URL>" "${OUTPUT_FILENAME}-${N}.xml" | sed
s/'^.*<URL><!\[CDATA\['//g | sed s/'\]\]\>\<\/URL\>'//g`

  N=`expr ${N} + 1`
  if [[ ${N} -lt 10 ]] ; then N=`echo "0"${N}` ; fi
  if [[ ${N} -gt 99 ]] ; then echo "bailing out after fetching the first
99 batches of data"; exit 0; fi

  echo "next url = ${NEXT_URL}"
  curl -H 'X-Requested-With: curl example QWEB 6.17' -k -u
"${USERNAME}:${PASS}" "${NEXT_URL}" > "${OUTPUT_FILENAME}-${N}.xml"
done
```

Excluded Hosts List

The Excluded Hosts List API v2 (**`api/2.0/fo/asset/excluded_ip/?action=list`**) allows API users to request a list of excluded hosts. The GET or POST access method may be used to make an API request.

Express Lite: This API is available to Express Lite users.

User Permissions

User permissions are described below.

User Role	Permissions
Manager	View all excluded hosts in subscription.
Auditor	View excluded compliance hosts in subscription.
Unit Manager	View excluded hosts in user's business unit.
Scanner	View excluded hosts in user's account.
Reader	View excluded hosts in user's account.

Input Parameters

The input parameters for the excluded hosts API are described below.

Parameter	Description
action=list	(Required) A flag used to make an excluded hosts list request.
echo_request={**0** | 1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When unspecified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
ips={value}	(Optional) Show only certain excluded IP addresses/ranges. When unspecified, all excluded IPs/ranges in your account will be listed. One or more IPs/ranges may be specified. Multiple entries are comma separated. An IP range is specified with a hyphen (for example, 10.10.24.1-10.10.24.20).
network_id={value}	(Optional, and valid only when the Network Support feature is enabled for the user's account) Restrict the request to a certain custom network ID.

Sample API Requests

These sample requests work on Qualys US Platform 1 where the FQDN in the API server URL is qualysapi.qualys.com. Please be sure to replace the FQDN with the proper API server URL for your platform. For the EU platform, use qualysapi.qualys.eu. For a partner platform, use the URL for your @customer platform API server.

The header parameter "X-Requested-With" is also provided as an example. Please change this parameter according to your need. This parameter is exposed in the activity log (only displayed when used with session cookie authentication - does not apply for basic authentication as used in the following examples) and it is useful for monitoring API activity.

Sample 1. Request an excluded hosts list that includes all excluded hosts in your account:

```
curl -s -u user:password -H 'X-Requested-With: curl demo 2'
-D headers.15 'https://qualysapi.qualys.com/api/2.0/fo/asset/excluded_ip/
?action=list'
```

Sample 2. Request an excluded hosts list that includes only IP addresses in the range 10.10.24.1-10.10.24.255:

```
curl -s -u user:password -H 'X-Requested-With: curl demo 2'
-D headers.16 'https://qualysapi.qualys.com/api/2.0/fo/asset/excluded_ip/
?action=list&ips=10.10.24.1-10.10.24.255'
```

XML Output

DTD for Excluded Hosts List

An excluded hosts API request returns XML output using the ip_list_output.dtd, which can be found at the following URL (where qualysapi.qualys.com is your API server URL):

```
https://qualysapi.qualys.com/api/2.0/fo/asset/excluded_ip/
ip_list_output.dtd
```

The DTD for the excluded hosts list XML is provided in Appendix B.

Sample Excluded Hosts List

Sample excluded hosts list XML output is shown below:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE IP_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/asset/excluded_ip/ip_list_output
.dtd">
```

```
<IP_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2015-04-23T00:33:24Z</DATETIME>
    <IP_SET>
      <IP_RANGE network_id="0" expiration_date="2015-04-
28T00:00:00Z">10.100.100.101-10.100.100.255</IP_RANGE>
      <IP network_id="14665885">10.10.10.1</IP>
      <IP network_id="0">10.100.100.100</IP>
      <IP network_id="0">100.100.100.100</IP>
    </IP_SET>
  </RESPONSE>
</IP_LIST_OUTPUT>
```

Excluded Hosts Change History

The excluded hosts change history V2 API (**`api/2.0/fo/asset/excluded_ip/history/?action=list`**) allows API users to request the change history for excluded hosts in the user's subscription. History record IDs in the XML output are listed in decreasing order. The GET or POST access method may be used to make an API request.

Unlike other V2 APIs, the excluded hosts change history returns change history records for all relevant IP addresses in the subscription, regardless of whether the user has access to these IP addresses in their account.

A maximum of 1,000 history records will be returned. If the maximum is reached, a warning message appears in the XML output with the URL to be used to obtain the next batch of records. The truncation warning is issued using the "id_max" parameter. See below for sample output showing this warning.

User Permissions

Users with these roles have permission to view all excluded hosts in the subscription: Manager, Auditor, Unit Manager, Scanner and Reader.

Input Parameters

The input parameters for the excluded hosts API are described below.

Parameter	Description
action=list	(Required) A flag used to show all changes made to excluded hosts within your subscription.
echo_request={**0**|1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When unspecified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
ips={value}	(Optional) Show only certain excluded IP addresses/ranges. When unspecified, all excluded IPs/ranges in your subscription will be listed. One or more IPs/ranges may be specified. Multiple entries are comma separated. An IP range is specified with a hyphen (for example, 10.10.24.1-10.10.24.20).
network_id={value}	(Optional, and valid only when the Network Support feature is enabled for the user's account) Restrict the request to a certain custom network ID.
id_min={value}	(Optional) Show only those history records in your subscription that have an ID number greater than or equal to an ID number you specify.

Parameter	Description
id_max={value}	(Optional) Show only those history records in your subscription that have an ID number less than or equal to an ID number you specify.
ids={value}	(Optional) Show only those history records in your subscription that have ID numbers matching the ID numbers you specify.

Sample API Requests

These sample requests work on Qualys US Platform 1 where the FQDN in the API server URL is qualysapi.qualys.com. Please be sure to replace the FQDN with the proper API server URL for your platform. For the EU platform, use qualysapi.qualys.eu. For a partner platform, use the URL for your @customer platform API server.

The header parameter "X-Requested-With" is also provided as an example. Please change this parameter according to your need. This parameter is exposed in the activity log (only displayed when used with session cookie authentication - does not apply for basic authentication as used in the following examples) and it is useful for monitoring API activity.

Sample 1. Request excluded hosts change history that includes all excluded hosts in your subscription:

```
curl -s -u user:password
-H 'X-Requested-With: curl demo 2' -D headers.15
'https://qualysapi.qualys.com/api/2.0/fo/asset/excluded_ip/history/?actio
n=list'
```

Sample 2. Request excluded hosts change history that includes only IP addresses in the range 10.10.24.1-10.10.24.255:

```
curl -s -u user:password
-H 'X-Requested-With: curl demo 2' -D headers.16
'https://qualysapi.qualys.com/api/2.0/fo/asset/excluded_ip/history/?actio
n=list&ips=10.10.24.1-10.10.24.255'
```

XML Output

DTD for Excluded Hosts Change History

An excluded hosts change history API request returns XML output using the history_list_output.dtd, which can be found at the following URL (where qualysapi.qualys.com is your API server URL):

```
https://qualysapi.qualys.com/api/2.0/fo/asset/excluded_ip/
history/history_list_output.dtd
```

The DTD for the excluded hosts change history XML is provided in Appendix B.

Sample Excluded Hosts Change History

Sample excluded hosts change history XML output is shown below:

```
<!DOCTYPE HISTORY_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/asset/excluded_ip/history/histor
y_list_output.dtd">

<HISTORY_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2011-03-18T01:48:42Z</DATETIME>
    <HISTORY_LIST>
      <HISTORY>
        <ID>1923</ID>
        <IP_SET>
          <IP_RANGE>10.10.10.2-10.10.10.11</IP_RANGE>
          <IP_RANGE>10.10.10.32-10.10.10.34</IP_RANGE>
          <IP>10.10.30.70</IP>
        </IP_SET>
        <ACTION>Added</ACTION>
        <DATETIME>2009-07-02T05:19:06Z</DATETIME>
        <USER_LOGIN>quays_ab</USER_LOGIN>
        <COMMENTS><![CDATA[DD]]></COMMENTS>
      </HISTORY>
      <HISTORY>
        <ID>1863</ID>
        <IP_SET>
          <IP_RANGE>10.10.10.102-10.10.10.120</IP_RANGE>
        </IP_SET>
        <ACTION>Removed</ACTION>
        <DATETIME>2009-06-01T23:51:26Z</DATETIME>
        <USER_LOGIN>quays_ab</USER_LOGIN>
        <COMMENTS><![CDATA[Removing 10.10.10.102-
10.10.10.120]]></COMMENTS>
      </HISTORY>
      <HISTORY>
        <ID>1663</ID>
        <IP_SET>
          <IP_RANGE>10.10.10.100-10.10.10.120</IP_RANGE>
        </IP_SET>
        <ACTION>Added</ACTION>
        <DATETIME>2009-04-29T06:56:13Z</DATETIME>
```

```
        <USER_LOGIN>quays_ss</USER_LOGIN>
        <COMMENTS><![CDATA[Scanner shouldn't add Exclude
hosts]]></COMMENTS>
      </HISTORY>

    ...

    </HISTORY_LIST>
    <WARNING>
      <CODE>1980</CODE>
      <TEXT>1,000 record limit exceeded. Use URL to get next batch of
results.</TEXT>
<URL><![CDATA[https://qualysapi.qualys.com/api/2.0/fo/asset/excluded_ip/h
istory/?action=list&id_max=1660]]></URL>
    </WARNING>
    <GLOSSARY>
      <USER_LIST>
        <USER>
          <USER_LOGIN>quays_ss</USER_LOGIN>
          <FIRST_NAME>Sally Unassigned</FIRST_NAME>
          <LAST_NAME>Storm</LAST_NAME>
          <ROLE>Scanner</ROLE>
        </USER>
        <USER>
          <USER_LOGIN>quays_ab</USER_LOGIN>
          <FIRST_NAME>Al</FIRST_NAME>
          <LAST_NAME>Berger</LAST_NAME>
          <ROLE>Manager</ROLE>
        </USER>
      </USER_LIST>
    </GLOSSARY>
  </RESPONSE>
</HISTORY_LIST_OUTPUT>
```

Manage Excluded Hosts

Manage your excluded IPs list using the Excluded IP API v2 (**`/api/2.0/fo/asset/excluded_ip/`**). The IPs in your excluded IPs list will not be scanned.

Add IPs

Use the parameter "action=add" to add IPs to your excluded IPs list.

Parameter	Description
action=add	(Required)
ips={value}	(Required) The IP addresses to be added to the excluded IPs list. Enter a comma separated list of IPv4 singletons or ranges. For example: 10.10.10.13,10.10.10.25-10.10.10.29
expiry_days={value}	(Optional) The number of days the IPs being added to the excluded IPs list will be considered valid for exclusion. When the expiration is reached, the IPs are removed from the list and made available again for scanning. When unspecified, the IPs being added have no expiration and will remain on the list until removed by a user.
dg_names={value}	(Optional) Specify users who will be notified 7 days before hosts are removed from the excluded hosts list (i.e. supply distribution group names as defined in the Qualys UI). Multiple distribution groups are comma separated. A maximum of 15 distribution groups may be entered.
comment={value}	(Required) User-defined notes (up to 1024 characters).
network_id={value}	(Optional and valid only when the user making the request has access to more than one network) Assign a network ID to the IPs being added to the excluded IPs list. By default, the user's default network ID is assigned.

API request 1 - add IPs:

```
curl -H "X-Requested-With: curl" -u "USERNAME:PASSWD" -d
"action=add&ips=10.100.100.101-10.100.100.255&comment=adding
ips&expiry_days=5"
"https://qualysapi.qualys.com/api/2.0/fo/asset/excluded_ip/"
```

XML output:

```
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
```

```
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2015-04-23T00:33:21Z</DATETIME>
    <TEXT>Adding IPs to Excluded IPs list.</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>Added IPs</KEY>
        <VALUE>10.100.100.101-10.100.100.255</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

API request 2 - IPs already in excluded IPs list:

```
curl -H "X-Requested-With: curl" -u "USERNAME:PASSWD" -d
"action=add&ips=10.10.34.210-10.10.34.212&comment=adding, added IPs "
"https://qualysapi.qualys.com/api/2.0/fo/asset/excluded_ip/"
```

XML output:

```
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2014-09-14T13:09:03Z</DATETIME>
    <TEXT>Not Adding any IPs to Excluded IPs list.</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>IPs already in Excluded IPs list.</KEY>
        <VALUE>10.10.34.210-10.10.34.212</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

Remove IPs

Use the parameter “action=remove” to remove IPs from your excluded IPs list.

Parameter	Description
action=remove	(Required)
ips={value}	(Required) The IP addresses to be removed from the excluded IPs list. Enter a comma separated list of IPv4 singletons or ranges. For example: 10.10.10.13,10.10.10.25-10.10.10.29

Parameter	Description
comment={value}	(Required) User-defined notes (up to 1024 characters).
network_id={value}	(Optional and valid only when the user making the request has access to more than one network) Identify a network ID that is assigned to the IPs being removed from the excluded IPs list. By default, the user's default network ID is assigned.

API request:

```
curl -H "X-Requested-With: curl" -u "USERNAME:PASSWD" -d
"action=remove&ips=10.10.34.250-10.10.34.254&comment=remove IPS"
"https://qualysapi.qualys.com/api/2.0/fo/asset/excluded_ip/"
```

XML output:

```
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2014-09-15T04:05:04Z</DATETIME>
    <TEXT>Removed IPs from Excluded IPs list.</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>Removed IPs</KEY>
        <VALUE>10.10.34.250-10.10.34.254</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

Remove All IPs

Use the parameter "action=remove_all" to remove all IPs from your excluded IPs list.

Parameter	Description
action=remove_all	(Required)
comment={value}	(Required) User-defined notes (up to 1024 characters).
network_id={value}	(Optional and valid only when the user making the request has access to more than one network) Identify a network ID that is assigned to the IPs being removed from the excluded IPs list. By default, the user's default network ID is assigned.

API request:

```
curl -H "X-Requested-With: curl" -u "USERNAME:PASSWD" -d
"action=remove_all&comment=remove all ips"
"https://qualysapi.qualys.com/api/2.0/fo/asset/excluded_ip/"
```

XML output:

```
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2015-04-24T00:08:19Z</DATETIME>
    <TEXT>Removed IPs from Excluded IPs list.</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>Removed IPs</KEY>
        <VALUE>10.100.100.101-10.100.100.255,100.100.100.101-
100.100.100.255</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

Purge Hosts

The Purge Hosts API (**`/api/2.0/fo/asset/host/?action=purge`**) is used to purge hosts. Purging hosts will remove automatic host data in the user's account (scan results will not be removed). Purged host information will not appear in new reports generated by users. The POST access method must be used to make an API request.

One or both types of host data is removed, based on the user's API request:

- Vulnerability data.
- Compliance data. Compliance scan data can only be available in a user's account when the policy compliance module is enabled for the subscription.

Optional input parameters allow users to purge hosts not scanned since a certain date and time. When the policy compliance module is enabled users have the option to purge compliance hosts not scanned since a certain date and time.

Express Lite: This API is available to Express Lite users.

User Permissions

User permissions are described below.

User Role	Permissions
Manager	Purge automatic host data for all hosts in the subscription, including vulnerability data and compliance data.
Auditor	Purge automatic host data for all compliance hosts in the subscription. Only compliance data will be removed (not vulnerability data).
Unit Manager Scanner Reader	Sub-accounts must be granted permissions. Purge vulnerability data: User must be granted the permission "Purge host information/history". When granted this permission, the users can purge vulnerability data for hosts in their account. Purge compliance data: User must be granted the permission "Purge host information/history" and the permission "Manage compliance". When granted these permissions, the user can purge compliance data for compliance hosts in their account.

Parameters

The parameters used to make a purge hosts request are described below.

Parameter	Description
action=purge	(Required) A flag used to make a purge hosts request.
echo_request={**0**|1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When unspecified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
ids={value}	(Optional) Purge host information for certain host IDs/ranges. One or more host IDs/ranges may be specified. Multiple entries are comma separated. A host ID range is specified with a hyphen (for example, 190-400).Valid host IDs are required.
	One of these host selection parameters must be specified in an API request: **ids**, **ips**, **ag_ids** or **ag_titles**. Multiple host selection parameters may be specified together in the same request.
ips={value}	(Optional) Purge host information certain IP addresses/ranges. One or more IPs/ranges may be specified. Multiple entries are comma separated. An IP range is specified with a hyphen (for example, 10.10.10.1-10.10.10.100).
	One of these host selection parameters must be specified in an API request: **ids**, **ips**, **ag_ids** or **ag_titles**. Multiple host selection parameters may be specified together in the same request.
ag_ids={value}	(Optional) Purge hosts belonging to asset groups with certain IDs. One or more asset group IDs and/or ranges may be specified. Multiple entries are comma separated. A range is specified with a dash (for example, 386941-386945). Valid asset group IDs are required.
	One of these host selection parameters must be specified in an API request: **ids**, **ips**, **ag_ids** or **ag_titles**. Multiple host selection parameters may be specified together in the same request. These parameters are mutually exclusive and cannot be specified together: **ag_ids** and **ag_titles**.

Parameter	Description
ag_titles={value}	(Optional) Purge hosts belonging to asset groups with certain strings in the asset group title. One or more asset group titles may be specified. Multiple entries are comma separated (for example, My+First+Asset+Group,Another+Asset+Group).
	One of these parameters must be specified in an API request: **`ids`**, **`ips`**, **`ag_ids`** or **`ag_titles`**. Multiple host selection parameters may be specified together in the same request. These parameters are mutually exclusive and cannot be specified together: **`ag_ids`** and **`ag_titles`**.
network_ids={value}	(Optional, and valid only when the Network Support feature is enabled for the user's account) Restrict the request to certain custom network IDs. Multiple network IDs are comma separated.
no_vm_scan_since={date}	(Optional) Purge hosts not scanned since a certain date and time (optional). The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT), like "2007-07-01" or "2007-01-25T23:12:00Z". User Permissions: An Auditor cannot be specify this parameter.
no_compliance_scan_since={date}	(Optional) Purge compliance hosts not scanned since a certain date and time (optional). This parameter is invalid for an Express Lite user. The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT), like "2007-07-01" or "2007-01-25T23:12:00Z". User Permissions: A sub-account (Unit Manager, Scanner or Reader) can specify this parameter only when the user account is granted certain permissions to purge compliance information. See "User Permissions".

Parameter	Description
compliance_enabled={**0**|1}	(Optional) This parameter is valid only when the policy compliance module is enabled for the user account. This parameter is invalid for an Express Lite user. Specify 1 to purge compliance hosts in the user's account. These hosts are assigned to the policy compliance module. When selected, the service will remove vulnerability information and compliance information associated with the selected hosts. Specify 0 to purge hosts which are not assigned to the policy compliance module. When selected, the service will remove vulnerability information associated with the selected hosts. User Permissions: A sub-account (Unit Manager, Scanner or Reader) can specify this parameter only when the user account is granted permissions to purge compliance information. See "User Permissions". An Auditor does not have permission to set **`compliance_enabled=0`**.
os_pattern={expression}	(Optional) Purge only hosts which have an operating system matching a certain regular expression. An empty value cannot be specified. Use "%5E%24" to match empty string. **Important Note:** The regular expression string you enter must follow the PCRE standard and it must be URL encoded. See page 149 for sample regular expression strings for matching operating system names. For information about the Perl Compatible Regular Expressions (PCRE) standard visit: http://php.net/manual/en/book.pcre.php For the PCRE syntax, see: http://php.net/manual/en/reference.pcre.pattern.syntax.php http://www.php.net/manual/en/reference.pcre.pattern.posix.php

XML Output

A purge hosts request returns XML output using the DTD "batch_return.dtd" which can be found at the following URL:

```
https://qualysapi.qualys.com/api/2.0/fo/batch_return.dtd
```

The DTD for the XML output is provided in Appendix A.

Virtual Host List

The virtual host list API (**`/api/2.0/fo/asset/vhost/?action=list`**) is used to view a list of virtual hosts in the user account. By default, all virtual hosts in the user account are included. Optional input parameters support filtering the list by port and IP address. The GET or POST access method may be used to make an API request.

Authentication is required for each API request. See Chapter 2, "Authentication Using the V2 APIs."

User permissions are described below.

User Role	Permissions
Manager	View virtual hosts in subscription.
Auditor	No permission to view virtual hosts.
Unit Manager	View virtual hosts for IP addresses in user's business unit.
Scanner	View virtual hosts for IP addresses in user's account.
Reader	View virtual hosts for IP addresses in user's account.

Parameters

The parameters used to make a virtual host list request are described below.

Parameter	Description
action=list	(Required) A flag used to make a virtual host list request.
echo_request={**0**|1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When unspecified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
ip={value}	(Optional) Show only virtual hosts that have a certain IP address.
port={value}	(Optional) Show only virtual hosts that have a certain port.

XML Output

A virtual host list request returns XML output using the DTD at the following URL:

```
https://qualysapi.qualys.com/api/2.0/fo/
asset/vhost/vhost_list_output.dtd
```

where `<qualysapi.qualys.com>` identifies the API server URL where your account is located. The DTD for the virtual host list XML is provided in Appendix B.

Take Actions on Virtual Hosts

The virtual host API (**/api/2.0/fo/asset/vhost/?action=<value>**) is used to create, edit and delete virtual hosts in the user account. One subscription can have a maximum of 1024 virtual hosts. The POST access method may be used to make an API request.

Authentication is required for each API request. See Chapter 2, "Authentication Using the V2 APIs."

User permissions are described below.

User Role	Permissions
Manager	Create, edit and delete virtual hosts in subscription.
Auditor	No permissions to create, edit and delete virtual hosts.
Unit Manager	Create, edit and delete virtual hosts for IP addresses in user's business unit. The "Create/edit virtual hosts" permission must be granted in the user's account.
Scanner	Create, edit and delete virtual hosts for IP addresses in user's account. The "Create/edit virtual hosts" permission must be granted in the user's account.
Reader	No permissions to create, edit and delete virtual hosts.

Parameters

The parameters used to make a request to create, edit or delete a virtual host request.

Parameter	Description
action={action}	(Required) A flag used to make a virtual host request: create (create a virtual host) update (update/edit a virtual host) delete (delete a virtual host) add_fqdn (add one or more FQDNs to a virtual host) delete_fqdn (remove one or more FQDNs from a virtual host)*
echo_request={**0**|1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When unspecified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
ip={value}	(Required) An IP address for the virtual host configuration.

Parameter	Description
port={value}	(Required) A port number for the virtual host configuration.
fqdn={value}	(Required for all actions except "delete". Invalid for "delete".) One or more fully-qualified domain names (FQDNs) for the virtual host configuration. Multiple entries are comma separated.*

* Note: A virtual host must have at least one FQDN. If you make a request using the parameter **`action=delete_FQDN`** it's not possible to remove all of the FQDNs defined for the virtual host.

XML Output

An API request to create, edit or delete a virtual hosts returns XML output using the "simple_return" DTD which can be found at the following URL:

```
https://qualysapi.qualys.com/api/2.0/simple_return.dtd
```

where `<qualysapi.qualys.com>` identifies the API server URL where your account is located.

The "simple return" DTD is provided in Appendix A.

Restricted IPs List

The Restricted IPs API v2 (**`/api/2.0/fo/setup/restricted_ips/`**) gives Managers (users assigned the Manager role) the ability to manage and update the list of restricted IPs within their subscription so this list stays in sync with their organization's security policy. Once the feature is activated and a restricted IPs list is enabled, only users logging in from restricted IPs will be allowed to connect to your Qualys subscription.

Parameters

Restricted IPs input parameters are described below.

Parameter	Description
action={value}	(Required) The action for the request. One of: activate - enable or disable the restricted IPs feature add - add restricted IPs delete - delete restricted IPs replace - replace restricted IPs list - download list of all restricted IPs clear - clear all restricted IPs and de-active this feature Allowed methods: For the actions activate, list and clear: GET or POST may be used. For the actions add, replace and delete: the POST method must be used when the "ips" parameter is specified, and the GET or POST method may be used when uploading CSV raw data.
echo_request={**0** | 1}	(Optional) Set to 1 if you want to include the input parameters in the XML output.
enable={**0** | 1}	(Optional; valid when action is activate) Enable or disable the restricted IPs list. Set enable=1 to enable the list; set enable=0 to clear any IPs in the list and disable the feature.
ips={value} -or- {CSV raw data upload}	(Optional; valid when action is add, replace or delete) The hosts you want to add to, remove from or replace in the restricted IPs list. IPs must be specified by using the "ips" parameter (using the POST method) or by uploading CSV raw data (using the GET or POST method). To upload CSV raw data using POST, specify --data-binary <data>. How to specify IP addresses. One or more IPs/ranges may be specified. Multiple IPs/ranges are comma separated. An IP range is specified with a hyphen (for example, 10.10.30.1-10.10.30.50). CIDR notation is supported.
output_format={CSV | **XML**}	(Optional; valid when action is list) The list output will be in XML format by default. For CSV format, set output_format=CSV.

XML Output

For XML output other than the restricted IPs list, simple return XML is used. The simple return DTD is described in Appendix A. The restricted IPs list output DTD is described in Appendix B.

Sample API Requests

Add Restricted IPs ("ips" parameter)

API Request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d
"action=add&ips=10.10.10.1-10.10.10.255"
"https://qualysapi.qualys.com/api/2.0/fo/setup/restricted_ips/" >
output.txt
```

XML Output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2013-03-22T11:04:52Z</DATETIME>
    <TEXT>Successfully added restricted ips</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>STATUS</KEY>
        <VALUE>disabled</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

Download Restricted IPs List

API Request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d
"action=list"
"https://qualysapi.qualys.com/api/2.0/fo/setup/restricted_ips/" >
output.txt
```

XML Output:

The DTD for the restricted IPs list XML is provided in Appendix B.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE RESTRICTED_IPS_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/setup/restricted_ips/restricted_
ips_output.dtd">
<RESTRICTED_IPS_OUTPUT>
  <RESPONSE>
    <DATETIME>2013-03-22T11:12:56Z</DATETIME>
    <IP_SET>
      <IP_RANGE>10.10.10.1-10.10.10.255</IP_RANGE>
    </IP_SET>
    <STATUS>disabled</STATUS>
  </RESPONSE>
</RESTRICTED_IPS_OUTPUT>
```

Replace Restricted IPs (IPs in CIDR notation)

API Request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d
"action=replace&ips=10.0.0.0/8"
"https://qualysapi.qualys.com/api/2.0/fo/setup/restricted_ips/" >
output.txt
```

XML Output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2013-03-22T11:45:00Z</DATETIME>
    <TEXT>Successfully replaced restricted ips</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>STATUS</KEY>
        <VALUE>disabled</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

Delete Restricted IPs (upload CSV raw data)

CSV raw data file (file1.csv):

```
$ cat file1.csv
10.0.0.1
10.0.0.2-10.0.0.100
```

API Request:

```
curl -H "X-Requested-with:curl" -H "Content-type:text/csv" -u
"USERNAME:PASSWORD" --data-binary "@file1.csv"
"https://qualysapi.qualys.com/api/2.0/fo/setup/restricted_ips/?action=del
ete"
```

XML Output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2013-03-22T11:45:34Z</DATETIME>
    <TEXT>Successfully deleted restricted ips</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>STATUS</KEY>
        <VALUE>disabled</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

Activate Restricted IPs feature and enable list

API Request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d
"action=activate&enable=1"
"https://qualysapi.qualys.com/api/2.0/fo/setup/restricted_ips/" >
output.txt
```

XML Output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2013-03-22T11:46:45Z</DATETIME>
    <TEXT>Restricted IPs feature has been enabled successfully</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>STATUS</KEY>
        <VALUE>enabled</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

Download Restricted IPs List in CSV format

API Request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d
"action=list&output_format=csv"
"https://qualysapi.qualys.com/api/2.0/fo/setup/restricted_ips/"
```

CSV Output:

```
----BEGIN_RESPONSE_BODY_CSV
10.0.0.0
10.0.0.101-10.255.255.255
----END_RESPONSE_BODY_CSV
----BEGIN_RESPONSE_FOOTER_CSV
STATUS
enabled
----END_RESPONSE_FOOTER_CSV
```

Clear All Restricted IPs and Disable the feature

API Request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d
"action=clear"
"https://qualysapi.qualys.com/api/2.0/fo/setup/restricted_ips/"
```

XML Output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2013-03-22T12:04:34Z</DATETIME>
    <TEXT>Successfully cleared restricted ips</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>STATUS</KEY>
        <VALUE>disabled</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

Add IPs

The Add Asset IP API v2 (**/api/2.0/fo/asset/ip/?action=add**) gives you the ability to add IP addresses to the subscription. Once added they are available for scanning. You can choose to add IP addresses to VM and/or PC, depending on your license.

Permissions: A Manager has permissions to add IP addresses. A Unit Manager can add IP addresses when the "Add assets" permission is enabled in their account. Users with other user roles (Scanner, Reader, Auditor) do not have permissions to add IP addresses.

Parameters

These input parameters are used to add IP addresses.

Parameter	Description
action=add	(Required) A flag used to make a request to add IP addresses to the subscription.
echo_request={**0**|1}	(Optional) Specify 1 to show (echo) the request's input parameters (names and values) in the XML output. When unspecified, input parameters are not included in the XML output.
ips={value} -or- {POSTed CSV raw data}	(Required) The hosts you want to add to the subscription. IPs must be specified by using the "ips" parameter (using the POST method) or by uploading CSV raw data (using the POST method). To upload CSV raw data, specify --data-binary <data>. How to specify IP addresses. One or more IPs/ranges may be specified. Multiple IPs/ranges are comma separated. An IP range is specified with a hyphen (for example, 10.10.30.1-10.10.30.50). CIDR notation is supported.
tracking_method={value}	(Optional) The tracking method is set to IP for IP address by default. To use another tracking method specify DNS or NETBIOS.
enable_vm={**0**|1} enable_pc={**0**|1}	(Required) You must enable the hosts for the VM application (enable_vm=1) or the PC application (enable_pc=1) or both VM and PC.
owner={value}	(Optional) The owner of the host asset(s). The owner must be a Manager or a Unit Manager. A valid Unit Manager must have the "Add assets" permission and sufficient remaining IPs (maximum number of IPs that can be added to the Unit Manager's business unit).

Parameter	Description
ud1={value} ud2={value} ud3={value}	(Optional) Values for user-defined fields 1, 2 and 3. You can specify a maximum of 128 characters (ascii) for each field value.
comment={value}	(Optional) User-defined comments.
ag_title={value}	(Required if the request is being made by a Unit Manager; otherwise invalid) The title of an asset group in the Unit Manager's business unit that the host(s) will be added to.

XML Output

The XML output uses the simple return DTD, described in Appendix A.

Sample API Request

API request (POSTED raw data in CSV format):

```
curl -H "X-Requested-With: Curl" -H "Content-Type:text/csv"
-u "USERNAME:PASSWORD" --data-binary @ips_list.csv
"https://qualysapi.qualys.com/api/2.0/fo/asset/ip/?action=add&enable_vm=1
&enable_pc=1&tracking_method=IP&owner=quays_es1"
```

API request ("ips" parameter):

```
curl -H "X-Requested-With: demo" -u "USERNAME:PASSWORD" -X "POST"
-d "action=add&enable_vm=1&enable_pc=1&ips=10.10.10.1,10.10.10.10-
10.10.10.20,10.10.10.200"
"https://qualysapi.qualys.com/api/2.0/fo/asset/ip/"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
  <!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
 <SIMPLE_RETURN>
    <RESPONSE>
      <DATETIME>2013-08-07T01:21:03Z</DATETIME>
      <TEXT>IPs successfully added to Vulnerability Management/Compliance
Management</TEXT>
    </RESPONSE>
</SIMPLE_RETURN>
```

Update IPs

The Update Asset IP API v2 (**`/api/2.0/fo/asset/ip/?action=update`**) gives you the ability to update IP addresses within the subscription.

Permissions: A Manager has permissions to update IP addresses. A Unit Manager can update IP addresses in asset groups assigned to the user's business unit. Users with other user roles (Scanner, Reader, Auditor) do not have permissions to update IP addresses.

Parameters

These input parameters are used to update IP addresses.

Parameter	Description
action=update	(Required) A flag used to make a request to update IP addresses within the subscription.
echo_request={**0** | 1}	(Optional) Specify 1 to show (echo) the request's input parameters (names and values) in the XML output. When unspecified, input parameters are not included in the XML output.
ips={value} -or- {POSTed CSV raw data}	(Required) The hosts within the subscription that you want to update. IPs must be specified by using the "ips" parameter (using the POST method) or by uploading CSV raw data (using the POST method). To upload CSV raw data, specify --data-binary <data>. How to specify IP addresses. One or more IPs/ranges may be specified. Multiple IPs/ranges are comma separated. An IP range is specified with a hyphen (for example, 10.10.30.1-10.10.30.50). CIDR notation is supported.
tracking_method={value}	(Optional) To use another tracking method specify IP for IP address, DNS or NETBIOS.
host_dns={**0** | 1} - or - host_netbios={**0** | 1}	(Optional and valid only when there is a single IP address specified in the IPs parameter) You can update the tracking method to DNS or NetBIOS if the update request is for a single IP address (host). To change to the DNS tracking method, specify host_dns=hostname, where hostname is the current DNS hostname for the IP address. To change to the NetBIOS tracking method, specify host_netbios=hostname, where hostname is the current NetBIOS hostname for the IP address.
	The parameter **`host_dns`** or **`host_netbios`** can be specified for an update request (both parameters can't be specified in the same request). The parameter **`tracking_method`** can't be specified in the same request with **`host_dns`** or **`host_netbios`**.

Parameter	Description
owner={value}	(Optional) The owner of the host asset(s). The owner must be a Manager. Another user (Unit Manager, Scanner, Reader) can be the owner if the IP address is in the user's account.
ud1={value} ud2={value} ud3={value}	(Optional) Values for user-defined fields 1, 2 and 3. You can specify a maximum of 128 characters (ascii) for each field value.
comment={value}	(Optional) User-defined comments.

XML Output

The XML output uses the simple return DTD, described in Appendix A.

Sample API Request

API request ("ips" parameter):

For the request below we're adding new IPs 10.10.10.200,10.10.23.40 (not already in the subscription) and assigning them the DNS tracking method.

```
curl -H "X-Requested-With: demo" -u "USERNAME:PASSWORD" -X "POST"
-d "action=update&ips=10.10.10.200,10.10.23.40&tracking_method=
DNS" "https://qualysapi.qualys.com/api/2.0/fo/asset/ip/"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2013-08-07T17:27:36Z</DATETIME>
    <TEXT>IPs successfully updated</TEXT>
  </RESPONSE>
</SIMPLE_RETURN>
```

Asset Group API

The Asset Group API v2 (**`/api/2.0/fo/asset/group/`**) lets you manage asset groups with more granularity. These operations are supported: list, add, edit and delete.

Permissions are below:

User Role	Permissions
Manager	- List all asset groups in the subscription. - Add, edit, delete asset groups in subscription.
Unit Manager	- List all asset groups in the user's business unit (those assigned to the business unit, and those owned by all users in the business unit). - Add, edit, delete asset groups owned by any user in the business unit.
Scanner	- List asset groups in the user's account (those assigned to the user, and those owned by the user). - Add, edit, delete asset groups owned by the user.
Reader	- List asset groups in the user's account (those assigned to the user). - No permission to add, edit, delete asset groups.

List all your asset groups

Use these parameters:

Parameter	Description
action=list	(Required) The GET or POST method may be used.
echo_request={**0**|1}	(Optional) Specify 1 to show (echo) the request's input parameters (names, values) in the XML output. When unspecified, parameters are not included in the XML output.
ids={value}	(Optional) Show only asset groups with certain IDs. Multiple IDs are comma separated.
id_min={value}	(Optional) Show only asset groups that have an ID greater than or equal to the specified ID.
id_max={value}	(Optional) Show only asset groups that have an ID less than or equal to the specified ID.

Parameter	Description
truncation_limit={value}	(Optional) Specify the maximum number of asset group records to output. By default this is set to 1000 records. If you specify truncation_limit=0, the output is not paginated and all records are returned in a single output. WARNING This can generate very large output and processing large XML files can consume a lot of resources on the client side. It is recommended to use the pagination logic and parallel processing. The previous page can be processed while the next page is being downloaded.
network_ids={value}	(Optional and valid only when the Networks feature is enabled in your account) Restrict the request to certain network IDs. Multiple IDs are comma separated.
unit_id={value}	(Optional) Show only asset groups that have a business unit ID equal to the specified ID.
user_id={value}	(Optional) Show only asset groups that have a user ID equal to the specified ID.
title={value}	(Optional) Show only the asset group that has a title equal to the specified string - this must be an exact match.
show_attributes={value}	(Optional) Show attributes for each asset group along with the ID. Your options are: None, All or a comma-separated list of attribute names. Attribute names: TITLE, OWNER, NETWORK_IDS, LAST_UPDATE, IP_SET, APPLIANCE_LIST, DOMAIN_LIST, DNS_LIST, NETBIOS_LIST, EC2_ID_LIST, HOST_IDS, USER_IDS, UNIT_IDS, BUSINESS_IMPACT, CVSS.

API request 1:

```
curl -u "USERNAME:PASSWD" -H "X-Requested-With: Curl" -X "POST" -d
"action=list&ids=442838"
"https://qualysapi.qualys.com/api/2.0/fo/asset/group/"
```

XML output 1:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE ASSET_GROUP_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/asset/group/asset_group_list_out
put.dtd">
<ASSET_GROUP_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2014-06-17T08:48:41Z</DATETIME>
    <ASSET_GROUP_LIST>
      <ASSET_GROUP>
        <ID>442838</ID>
```

```
        <TITLE><![CDATA[All]]></TITLE>
        <OWNER_ID>103448</OWNER_ID>
        <UNIT_ID>0</UNIT_ID>
        <NETWORK_ID>0</NETWORK_ID>
        <IP_SET>
          <IP_RANGE>10.10.10.0-10.10.10.1</IP_RANGE>
          <IP_RANGE>10.10.10.3-10.10.10.6</IP_RANGE>
          <IP>10.10.10.14</IP>
          <IP_RANGE>10.10.10.16-10.10.10.20</IP_RANGE>
          <IP_RANGE>10.10.10.22-10.10.10.255</IP_RANGE>
          <IP>10.10.31.26</IP>
        </IP_SET>
      </ASSET_GROUP>
    </ASSET_GROUP_LIST>
  </RESPONSE>
</ASSET_GROUP_LIST_OUTPUT>
```

API request 2 - show all attributes:

```
curl -u "USERNAME:PASSWD" -H "X-Requested-With: Curl" -X "POST" -d
"action=list&ids=634851&show_attributes=ALL"
"https://qualysapi.qualys.com/api/2.0/fo/asset/group/"
```

XML output 2:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE ASSET_GROUP_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/asset/group/asset_group_list_out
put.dtd">
<ASSET_GROUP_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2014-06-17T09:08:01Z</DATETIME>
    <ASSET_GROUP_LIST>
      <ASSET_GROUP>
        <ID>634851</ID>
        <TITLE><![CDATA[mp1]]></TITLE>
        <OWNER_ID>70953</OWNER_ID>
        <UNIT_ID>0</UNIT_ID>
        <LAST_UPDATE>2014-06-13T11:44:04Z</LAST_UPDATE>
        <BUSINESS_IMPACT>High</BUSINESS_IMPACT>
        <CVSS_ENVIRO_CDP>Not Defined</CVSS_ENVIRO_CDP>
        <CVSS_ENVIRO_TD>Not Defined</CVSS_ENVIRO_TD>
        <CVSS_ENVIRO_CR>Not Defined</CVSS_ENVIRO_CR>
        <CVSS_ENVIRO_IR>Not Defined</CVSS_ENVIRO_IR>
        <CVSS_ENVIRO_AR>Not Defined</CVSS_ENVIRO_AR>
        <DEFAULT_APPLIANCE_ID>43575</DEFAULT_APPLIANCE_ID>
        <APPLIANCE_IDS>43576, 43575</APPLIANCE_IDS>
        <DOMAIN_LIST>
          <DOMAIN netblock="10.10.10.0, 10.10.25.50">ad.lan</DOMAIN>
```

```
        </DOMAIN_LIST>
        <NETBIOS_LIST>
          <NETBIOS>WIN2003-SRV-O</NETBIOS>
        </NETBIOS_LIST>
      </ASSET_GROUP>
    </ASSET_GROUP_LIST>
  </RESPONSE>
</ASSET_GROUP_LIST_OUTPUT>
```

DTD:

The DTD for the asset group list output can be found at this URL (where <qualysapi.qualys.com> is the API server URL where your account is located):

```
https://qualysapi.qualys.com/api/2.0/fo/asset/group/asset_group_list_outp
ut.dtd
```

See Appendix B for details.

Add a new asset group

Use these parameters:

Parameter	Description
action=add	(Required) The POST method must be used.
echo_request={**0**|1}	(Optional) Specify 1 to show (echo) the request's input parameters (names, values) in the XML output. When unspecified, parameters are not included in the XML output.
title={value}	(Required) An asset group title. This name must be unique and can't be "All".
network_id={value}	(Optional) The network ID of the network you want to assign the asset group to.
See "Asset Group Parameters"	

API request:

```
curl -u "USERNAME:PASSWD" -H "X-Requested-With: Curl" -X "POST"
-d "title=MY DEMO AG&network_id=1220&comments=This is
comment&division=this is divison&location=this is
location&business_impact=high&cvss_enviro_cdp=low&cvss_enviro_td=low&cvss
_enviro_cr=medium&cvss_enviro_ir=high&cvss_enviro_ar=medium&ips=10.1.1.1/
31" "https://qualysapi.qualys.com/api/2.0/fo/asset/group/?action=add"
```

XML output:

```
?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2014-05-28T22:57:50Z</DATETIME>
    <TEXT>Asset Group successfully added.</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>395752377</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

Edit an asset group

Use these parameters:

Parameter	Description
action=edit	(Required) The POST method must be used.
echo_request={**0** | 1}	(Optional) Specify 1 to show (echo) the request's input parameters (names, values) in the XML output. When unspecified, parameters are not included in the XML output.
id={value}	(Required) The ID of the asset group you want to edit.
See "Asset Group Parameters"	

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST"
-d "id=395752377&set_title=MY ASSET GROUP"
"https://qualysapi.qualys.com/api/2.0/fo/asset/group/?action=edit"
```

XML output:

The XML output uses the simple return (/api/2.0/simple_return.dtd).

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2014-05-29T15:29:00Z</DATETIME>
```

```
    <TEXT>Asset Group Updated Successfully</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>395752377</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

Delete an asset group

By deleting an asset group any scheduled scans using the asset group will be deactivated.

Use these parameters:

Parameter	Description
action=delete	(Required) The POST method must be used.
echo_request={**0**|1}	(Optional) Specify 1 to show (echo) the request's input parameters (names, values) in the XML output. When unspecified, parameters are not included in the XML output.
id={value}	(Required) The ID of the asset group you want to delete.

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST"
-d "id=395752377"
"https://qualysapi.qualys.com/api/2.0/fo/asset/group/?action=delete"
```

XML output:

The XML output uses the simple return (/api/2.0/simple_return.dtd).

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2014-05-29T15:49:35Z</DATETIME>
    <TEXT>Asset Group Deleted Successfully</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>395752377</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
```

```
</SIMPLE_RETURN>
```

Asset Group Parameters

Theses parameters are used for adding and editing an asset group.

IPs will be silently ignored when 1) the IPs are not in your account, or 2) the IPs are not in the network assigned to the asset group - when the Networks feature is enabled. This same logic applies to domains, appliances, DNS names and NetBIOS names.

The "set" (overwrite) and "remove" operations can cause the asset group to have no IPs, domains, etc depending on the parameter.

Parameter	Parameter Name action=add	Parameter Name action=edit
Comments	comments (255 characters maximum)	set_comments
Division	division (64 characters maximum)	set_division
Function	function (64 characters maximum)	set_function
Location	location (64 characters maximum)	set_location
Business Impact	business_impact (One of: critical, high, medium, low, none)	set_business_impact
IP addresses/ranges	ips	add_ips remove_ips set_ips
Scanner Appliances	appliance_ids	add_appliance_ids remove_appliance_ids set_appliance_ids
Default Scanner Appliance	default_appliance_id	set_default_appliance_id
Domains	domains	add_domains remove_domains set_domains
DNS Names	dns_names	add_dns_names remove_dns_names set_dns_names

Parameter	Parameter Name action=add	Parameter Name action=edit
NetBIOS Names	netbios_names	add_netbios_names remove_netbios_names set_netbios_names
Title	title (255 characters maximum)	set_title
CVSS Environmental Metric: Collateral Damage Potential	cvss_enviro_cdp (One of: high, medium-high, low-medium, low, none)	set_cvss_enviro_cdp
CVSS Environmental Metric: Target Distribution	cvss_enviro_td (One of: high, medium, low, none)	set_cvss_enviro_td
CVSS Environmental Metric: Confidentiality Requirement	cvss_enviro_cr (One of: high, medium, low)	set_cvss_enviro_cr
CVSS Environmental Metric: Integrity Requirement	cvss_enviro_ir (One of: high, medium, low)	set_cvss_enviro_ir
CVSS Environmental Metric: Availability Requirement	cvss_enviro_ar (One of: high, medium, low)	set_cvss_enviro_ar

CHAPTER 6

IPv6 Asset API

The IPv6 Asset API allows Manager users to manage IPv6 assets so they can be scanned using Qualys. The IPv6 API can be used when the IPv6 Support feature is enabled in the user's subscription. Please contact Support if you would like this feature enabled for your account.

This chapter describes how to use the IPv6 Asset API, built on the API V2 Architecture.

These topics are covered:

- API Support for IPv6 Asset Management and Scanning
- View IPv6 Mapping Records
- Add IPv6 Mapping Records
- Remove IPv6 Mapping Records

API Support for IPv6 Asset Management and Scanning

IPv6 Support is a subscription-level option that must be enabled for your subscription by Qualys Support in order to start managing and scanning IPv6 hosts. Follow the steps below to get started with managing and scanning IPv6 hosts using the API.

Step 1: Add Special IPv4 Addresses to your subscription

Using the Asset API add to your subscription the special, mapping IPv4 addresses. These IPv4 addresses are used for mapping IPv4 addresses to your IPv6 hosts. The IPv4 addresses for mapping are in the special 0.0.0.0/8 network, in this range:

```
0.0.0.1-0.254.255.255
```

A sample request for adding the special IPv4 addresses is shown below (where qualysapi.qualys.com is the server URL where your Qualys account is located):

```
https://qualysapi.qualys.com/msp/asset_ip.php?action=add&
host_ips=0.0.0.1-0.0.0.255
```

Step 2: Add IPv6 Mapping Records

Manager users can add and remove IPv6 mapping records for the subscription by submitting the records in CSV or XML format. Each mapping record associates one IPv6 address in your network to one IPv4 address in the special mapping range 0.0.0.1-0.254.255.255. A maximum of 10,000 records can be added or removed per API request.

How to Add IPv6 Records in CSV

Review the steps below to learn how to add IPv6 mapping records by submitting the records in CSV format. A curl client is used to illustrate this process.

1) View Mapping Records in CSV

Input:

```
$ curl -u username:password -H "X-Requested-With: curl"
"https://qualysguard.api.qualys.com/api/2.0/fo/asset/ip/v4_v6/?act
ion=list&output_format=csv"
```

Output:

Note: The service automatically returns an ID value in the ID column for each IPv6 mapping record. This ID is assigned by the service when the record is created.

```
----BEGIN_RESPONSE_BODY_CSV
ID,IPv4,IPv6
"46947","0.0.0.7","2001:db8:85a3::8a2e:370:84"^M
"47036","0.0.0.1","2001:db8:85a3::8a2e:370:77"^M
----END_RESPONSE_BODY_CSV
----BEGIN_RESPONSE_FOOTER_CSV
"Status Message"^M
"Finished"^M
----END_RESPONSE_FOOTER_CSV
```

2) Prepare file1.csv with records to be added

The CSV file contents identify one or more IPv6 mapping records to be added. The columns in the CSV upload file are described below.

Column	Description
IPv4	(Required) An IPv4 address. The IPv4 address can be defined in only one IPv6 mapping record within your subscription.
IPv6	(Required) An IPv6 address. The IPv6 address can be defined in only one IPv6 mapping record within your subscription.
ID	(Optional) A user-defined, custom ID may be included. **Important:** Custom ID values will not be saved with record data within your subscription.

The CSV file must include the input parameters **`action=add`** and **`csv_data=`**. The parameter **`all_or_nothing`** is optional. When set to 1 or unspecified, the service cancels the request and does not add any new records if it finds the upload data has one record with an IP conflict. When set to 0 the service does not cancel the request if an IP conflict is found.

Sample file1.csv used to add IPv6 mapping records:

```
$ cat file1.csv
action=add&all_or_nothing=1&csv_data=
"0.0.0.2","2001:470:8418:a18::a0a:1805"%0A
"0.0.0.3","2001:470:8418:a18::a0a:ab7"%0A
"0.0.0.4","2001:470:8418:a18::a0a:1849"%0A
"0.0.0.5","2001:470:8418:a18::a0a:189c"%0A
"0.0.0.6","2001:470:8418:a18::a0a:189d"%0A
"0.0.0.8","2001:470:8418:a18::a0a:189e"%0A
"0.0.0.9","2001:470:8418:a18::a0a:18d0"%0A
"0.0.0.10","2001:470:8418:a18::a0a:18d1"%0A
"0.0.0.11","2001:470:8418:a18::a0a:18d2"%0A
```

```
"0.0.0.12","2001:470:8418:a18::a0a:18d6"%0A
"0.0.0.13","2001:470:8418:a18::a0a:18d7"%0A
"0.0.0.14","2001:470:8418:a18::a0a:18da"%0A
"0.0.0.15","2001:470:8418:a18::a0a:18db"%0A
"0.0.0.16","ff00:abcd::1234"%0A
```

3) POST data from file1.csv (Success)

Input:

```
$ curl -u username:password -H "X-Requested-With: curl"
-d @file1.csv
"https://qualysguard.api.qualys.com/api/2.0/fo/asset/ip/v4_v6/"
```

Output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysguard.api.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2011-11-03T19:31:27Z</DATETIME>
    <TEXT>Successfully imported 14 records
</TEXT>
  </RESPONSE>
</SIMPLE_RETURN>
```

How to Add IPv6 Records in XML

Review the steps below to learn how to add IPv6 mapping records by submitting the records in XML format. A curl client is used to illustrate this process.

1) View mapping records in XML

Input:

```
$ curl -u username:password -H "X-Requested-With: curl"
"https://qualysguard.api.qualys.com/api/2.0/fo/asset/ip/v4_v6/?act
ion=list&output_format=xml"
```

Output:

Note: The service automatically returns an ID value in the <ID> element for each IPv6 mapping record. This ID is assigned by the service when the record is created.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE IP_MAP_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/asset/ip/v4_v6/ip_map_lis
t_output.dtd">
<IP_MAP_LIST_OUTPUT>
 <RESPONSE>
   <DATETIME>2011-11-28T19:42:10Z</DATETIME>
   <IP_MAP_LIST>
     <IP_MAP>
       <ID>46947</ID>
       <V4>0.0.0.7</V4>
       <V6>2001:db8:85a3::8a2e:370:84</V6>
     </IP_MAP>
     <IP_MAP>
       <ID>47036</ID>
       <V4>0.0.0.1</V4>
       <V6>2001:db8:85a3::8a2e:370:77</V6>
     </IP_MAP>
   </IP_MAP_LIST>
 </RESPONSE>
</IP_MAP_LIST_OUTPUT>
```

2) Prepare file2.xml with records to be added

The XML file contents identify one or more IPv6 mapping records to be added. The element in the XML upload file are described below.

Column	Description
<V4>	(Required) An IPv4 address. The IPv4 address can be defined in only one IPv6 mapping record within your subscription.
<V6>	(Required) An IPv6 address. The IPv6 address can be defined in only one IPv6 mapping record within your subscription.
<ID>	(Optional) A user-defined, custom ID may be included. **Important:** Custom ID values will not be saved with record data within your subscription.

The XML file must include the input parameters **`action=add`** and **`xml_data=`**. The parameter **`all_or_nothing`** is optional. When set to 1 or unspecified, the service cancels the request and does not add any new records if it finds the upload data has one record with an IP conflict. When set to 0 the service does not cancel the request if an IP conflict is found.

Sample file2.xml used to add IPv6 mapping records:

```
$ cat file2.xml
action=add&xml_data=
<IP_MAP_LIST>
  <IP_MAP>
    <V4>0.0.0.2</V4>
    <V6>2001:470:8418:a18::a0a:1805</V6>
  </IP_MAP>
  <IP_MAP>
    <V4>0.0.0.3</V4>
    <V6>2001:470:8418:a18::a0a:ab7</V6>
  </IP_MAP>
</IP_MAP_LIST>
```

3) POST data from file2.xml (Success)

Input:

```
$ curl -u username:password -H "X-Requested-With: curl"
-d @file2.xml
"https://qualysguard.api.qualys.com/api/2.0/fo/asset/ip/v4_v6/"
```

Output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysguard.api.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2011-11-03T20:59:07Z</DATETIME>
    <TEXT>Successfully imported 2 records</TEXT>
  </RESPONSE>
</SIMPLE_RETURN>
```

Step 3: Remove IPv6 Mapping Records (optional)

Manager users can remove IPv6 mapping records for the subscription by submitting the records to be removed in CSV or XML format. A maximum of 10,000 records can be removed per API request.

It's not necessary to specify both the IPv4 address and the IPv6 address for each record to be deleted in the data file (CSV or XML). If you specify only the IPv4 address, any associated record will be deleted. If you specify only the IPv6 address, any associated record will be deleted. If you specify both the IPv4 and IPv6 addresses, any record containing either address will be deleted. If no IP addresses specified in a mapping record to be deleted match any IP addresses already defined in mapping records in the subscription, the mapping record listed in the data file will be silently ignored.

Important: When an IPv6 mapping record is removed, any scan data associated with your IPv6 host is removed from your subscription and this data is not recoverable.

How to Remove IPv6 Records in CSV

Review the steps below to learn how to remove IPv6 mapping records by submitting the records in CSV format. A curl client is used to illustrate this process.

1) View mapping records in CSV

Input:

```
$ curl -u username:password -H "X-Requested-With: curl"
"https://qualysguard.api.qualys.com/api/2.0/fo/asset/ip/v4_v6/?act
ion=list&output_format=csv"
```

2) Prepare file3.csv with records to be removed

The CSV file contents identify one or more IPv6 mapping records to be removed.

Sample file3.csv used to remove IPv6 mapping records:

```
$ cat file3.csv
action=remove&csv_data=
"0.0.0.4","2001:470:8418:a18::a0a:1849"%0A
"0.0.0.5","2001:470:8418:a18::a0a:189c"%0A
```

3) POST data from file3.csv (Success)

Input:

```
$ curl -u username:password -H "X-Requested-With: curl"
-d @file3.csv
"https://qualysguard.api.qualys.com/api/2.0/fo/asset/ip/v4_v6/"
```

Output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
```

```
"https://qualysguard.api.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2011-11-03T19:31:27Z</DATETIME>
    <TEXT>Removed 2 records (any associated scanned host data is
now queued for purging)</TEXT>
  </RESPONSE>
</SIMPLE_RETURN>
```

How to Remove IPv6 Records in XML

Review the steps below to learn how to remove IPv6 mapping records by submitting the records in XML format. A curl client is used to illustrate this process.

1) View mapping records in XML

Input:

```
$ curl -u username:password -H "X-Requested-With: curl"
"https://qualysguard.api.qualys.com/api/2.0/fo/asset/ip/v4_v6/?act
ion=list&output_format=xml"
```

2) Prepare file4.xml with records to be removed

The XML file contents identify one or more IPv6 mapping records to be removed.

Sample file4.XML used to remove IPv6 mapping records:

```
$ cat file4.xml
action=remove&xml_data=
<IP_MAP_LIST>
  <IP_MAP>
    <V4>0.0.0.4</V4>
    <V6>2001:470:8418:a18::a0a:1849</V6>
  </IP_MAP>
  <IP_MAP>
    <V4>0.0.0.5</V4>
    <V6>2001:470:8418:a18::a0a:189c</V6>
  </IP_MAP>
</IP_MAP_LIST>
```

3) POST data from file4.xml (Success)

Input:

```
$ curl -u username:password -H "X-Requested-With: curl"
-d @file4.xml
"https://qualysguard.api.qualys.com/api/2.0/fo/asset/ip/v4_v6/"
```

Output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysguard.api.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2011-11-03T20:59:07Z</DATETIME>
    <TEXT>Removed 2 records (any associated scanned host data is
now queued for purging)</TEXT>
  </RESPONSE>
</SIMPLE_RETURN>
```

Step 4: Enable IPv6 for Scanner Appliance(s)

IPv6 scanning is supported using a scanner appliance enabled with IPv6. You can enable this by editing the appliance within the Qualys user interface. Once IPv6 is enabled, the appliance uses stateless address autoconfiguration to obtain an IPv6 address from the router (note that stateful configuration through DHCPv6 or Static IPv6 is not supported).

Step 5: Launch Scan

Using the Qualys API you can launch scans on the IPv4 addresses which are mapped to IPv6 addresses.

Step 6: View IPv6 Addresses using Host List Detection API

The scan results XML output will include IPv4 addresses only. Also, scan reports downloaded from the user interface will include IPv4 addresses only.

The host list detection output returned from a host list detection API request (**`api/2.0/fo/asset/host/vm/detection/?action=list`**) gives you the IPv6 address, if available, along with the "automatic" vulnerability detection data.

To request a list of VM scanned hosts which have IPv4 addresses that are mapped to IPv6 addresses in your account, you enter the IPv4 addresses for the **`ips`** parameter.

For example, if the special IPv4 address 0.0.0.199 is mapped to an IPv6 address in your account and this IP address has been scanned, you can make this API request:

```
curl -H 'X-Requested-With: Curl Sample' -u 'username:password'
'https://qualysapi.qualys.com/api/2.0/fo/asset/host/vm/detectio
n/?action=list&ips=0.0.0.100'
```

XML output returned will show the IPv4 address and the IPv6 address for the host, as shown below (XML fragment):

```
...
<HOST>
        <ID>276010</ID>
        <IP>0.0.0.100</IP>
        <IPV6>2001:470:8418:a18::a0a:18c7</IPV6>
        <TRACKING_METHOD>IP</TRACKING_METHOD>
        <OS><![CDATA[Windows 2003 Service Pack 2]]></OS>
        <DNS><![CDATA[mssql2k8-24-
199.patch.ad.vuln.qa.qualys.com]]></DNS>
        <LAST_SCAN_DATETIME>2010-11-
17T19:06:31Z</LAST_SCAN_DATETIME>
        <DETECTION_LIST>
        ...
```

View IPv6 Mapping Records

The **/api/2.0/fo/asset/ip/v4_6** resource with the **action=list** parameter is used to view a list of IPv6 mapping records. Each mapping record associates one IPv6 address in your network with one IPv4 address in the special mapping range 0.0.0.1-0.254.255.255. The GET method may be used to make an API request. Authentication is required; see Chapter 2, "Authentication Using the V2 APIs."

A maximum of 5,000 IPv6 mapping records will be processed per request, unless the **truncation_limit** input parameter is specified. If the requested list identifies more than 5,000 records or the number of records specified using **truncation_limit**, then the XML output includes the <WARNING> element and instructions for making another request for the next batch of records.

Permissions

User permissions are described below.

User Role	Permissions
Manager	View all IPv6 mapping records, when the IPv6 Support feature is enabled for the user's subscription.
Auditor, Unit Manager, Scanner, Reader	No permission to view all IPv6 mapping records.

Parameters

The input parameters used to request a list of IPv6 mapping records are described below.

Parameter	Description
action=list	(Required) The action type for requesting a list of IPv6 mapping records.
echo_request={**0** | 1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
id_min={value}	(Optional) Show only mapping records which have a minimum record ID. A valid mapping record ID is required. When unspecified, records are not filtered by record ID.
id_max={value}	(Optional) Show only mapping records which have a maximum record ID. A valid mapping record ID is required.
ipv4_filter={value}	(Optional) Show only mapping records with certain IPv4 addresses. When unspecified, records are not filtered by IPv4 addresses.

Parameter	Description
ipv6_network={value}	(Optional) Show only mapping records with certain IPv6 network addresses. When unspecified, records are not filtered by IPv6 network addresses.
output_format={**CSV** | XML}	(Optional) The requested output format: CSV or XML. When unspecified, the output format will be CSV. Note: When the service outputs CSV, each line ends with a carriage-return and linefeed pair (ASCII/CRLF=0x0D 0x0A).
truncation_limit={value}	(Optional) The maximum number of mapping records to be returned by the API request. A valid value is an integer between 1 and 1,000,000. When unspecified, 5,000 records will be returned.

XML Output

DTD for IPv6 Mapping Records List

An API request for a mapping record list returns XML output using the DTD which can be found at the following URL:

```
https://qualysapi.qualys.com/api/2.0/fo/asset/ip/v4_v6/
asset/ip/v4_v6/ip_map_list_output.dtd
```

The DTD for the IPv6 mapping records list XML is provided in Appendix B.

Sample IPv6 Mapping Records List Output

For sample mapping records list output in CSV format, see "How to Add IPv6 Records in CSV."

For sample mapping records list output in XML format, see "How to Add IPv6 Records in XML."

Add IPv6 Mapping Records

The **/api/2.0/fo/asset/ip/v4_6** resource with the **action=add** parameter is used to add IPv6 mapping records to the subscription. Each mapping record associates one IPv6 address in your network with one IPv4 address in the special mapping range 0.0.0.1-0.254.255.255. The POST method may be used to make an API request. Authentication is required; see Chapter 2, "Authentication Using the V2 APIs."

A maximum of 10,000 mapping records can be added per API request.

Permissions

User Role	Permissions
Manager	Add IPv6 mapping records, when the IPv6 Support feature is enabled for the user's subscription.
Auditor, Unit Manager, Scanner, Reader	No permission to add IPv6 mapping records.

Parameters

The input parameters used to add IPv6 mapping records are described below.

Parameter	Description
action=add	(Required) The action type to add IPv6 mapping records.
echo_request={**0** | 1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
csv_data={value}	The CSV data file containing the IPv6 mapping records that you want to add. This parameter or **xml_data** must be specified. See "How to Add IPv6 Records in CSV."
	The parameters **csv_data** and **xml_data** cannot be specified in the same request.

Parameter	Description
xml_data={value}	The CSV data file containing the IPv6 mapping records that you want to add. This parameter or **csv_data** must be specified. See "How to Add IPv6 Records in XML."
	The parameters **csv_data** and **xml_data** cannot be specified in the same request.
all_or_nothing={0|**1**}	(Optional) This parameter controls how the service processes the IPv6 mapping records in the upload data. When unspecified or set to **1**, the service cancels the request and does not add any new records once it finds the upload data has one record with an IP conflict. When set to **0** the service does not cancel the request if an IP conflict is found.

XML Output

DTD for Simple Return

An API request to add IPv6 mapping records returns XML output using the simple return DTD which can be found at the following URL:

```
https://qualysapi.qualys.com/api/2.0/simple_return.dtd
```

A description of the simple return DTD is provided in Appendix A.

Sample XML Output

For simple return output in CSV format, see "How to Add IPv6 Records in CSV."

For simple return output in XML format, see "How to Add IPv6 Records in XML."

Remove IPv6 Mapping Records

The **`/api/2.0/fo/asset/ip/v4_6`** resource with the **`action=remove`** parameter is used to remove IPv6 mapping records from the subscription. The POST method may be used to make an API request. Authentication is required; see Chapter 2, "Authentication Using the V2 APIs."

A maximum of 10,000 mapping records can be removed per API request.

It's not necessary to specify both the IPv4 address and the IPv6 address for each record to be deleted in the data file (CSV or XML). If you specify only the IPv4 address, any associated record will be deleted. If you specify only the IPv6 address, any associated record will be deleted. If you specify both the IPv4 and IPv6 addresses, any record containing either address will be deleted. If no IP addresses specified in a mapping record to be deleted match any IP addresses already defined in mapping records in the subscription, the mapping record listed in the data file will be silently ignored.

Important: When an IPv6 mapping record is removed, any scan data associated with your IPv6 host is removed from your subscription and this data is not recoverable.

Permissions

User Role	Permissions
Manager	Remove all IPv6 mapping records, when the IPv6 Support feature is enabled for the user's subscription.
Auditor, Unit Manager, Scanner, Reader	No permission to remove IPv6 mapping records.

Parameters

The input parameters used to remove IPv6 mapping records are described below.

Parameter	Description
action=remove	(Required) The action type to remove IPv6 mapping records.
echo_request={**0**|1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.

Parameter	Description
csv_data={value}	The CSV data file containing the IPv6 mapping records that you want to remove from your subscription. This parameter or **xml_data** must be specified. See "How to Remove IPv6 Records in CSV."
xml_data={value}	The CSV data file containing the IPv6 mapping records that you want to remove from your subscription. This parameter or **csv_data** must be specified. See "How to Remove IPv6 Records in XML."

XML Output

DTD for Simple Return

An API request to remove IPv4 to IPv6 mapping records returns XML output using the simple return DTD which can be found at the following URL:

```
https://qualysapi.qualys.com/api/2.0/simple_return.dtd
```

A description of the simple return DTD is provided in Appendix A.

Sample XML Output

For simple return output in CSV format, see "How to Remove IPv6 Records in CSV."

For simple return output in XML format, see "How to Remove IPv6 Records in XML."

CHAPTER 7

Compliance API

Qualys Policy Compliance allows customers to audit host configurations and measure their level of compliance with internal and external policies. The Compliance API allows API users to report on policy compliance data in their user account.

This chapter describes how to use the Compliance API, built on the API V2 Architecture.

These topics are covered:

- Compliance Control List
- Compliance Policy List
- Compliance Policy - Import and Export
- Compliance Policy - Merge
- Compliance Policy - Manage Asset Groups
- Compliance Posture Information
- Control Criticality
- SCAP Cyberscope Report
- SCAP ARF Report
- SCAP Policy List

Compliance Control List

The "Compliance Control List" API v2 (the resource **`/api/2.0/fo/compliance/control/`** with the parameter **`action=list`**) is used to view a list of compliance controls which are visible to the user. The compliance control ID for each control is listed in the output. Controls in the XML output are sorted by control ID in ascending order. Optional input parameters support filtering the list. Authentication is required for each API request. See Chapter 2, "Authentication Using the V2 APIs."

The user has the ability to select the amount of additional information to include for each control in the output. By default, this basic control information is included: the control ID, the control category, the control sub-category, the control statement, and a list of technologies.

Use the **`details`** input parameter to select another level of detail to be included in the control list output. Two parameter settings are available. Specify **`details=All`** to show the basic compliance control information and a list of framework mappings for each control. Specify **`details=None`** to list the compliance control ID only for each control.

Using the Qualys user interface, it's possible to customize the list of frameworks at the subscription level. Under PC, go to Policies > Setup > Frameworks to customize the frameworks list. If the frameworks list is customized for your subscription, then the customized list of frameworks will appear in the controls list output returned by a control list API request.

Maximum Controls per API Request

The output of the Compliance Control API is paginated. By default, a maximum of 1,000 control records are returned per request. You can customize the page size (i.e. the number of control records) by using the parameter "truncation_limit=2000" for instance. In this case the results will be return with pages of 2,000 records.

User Permissions

User permissions are described below.

User Role	Permissions
Manager	View all compliance controls.
Auditor	View all compliance controls.
Unit Manager	View all compliance controls.
Scanner	View all compliance controls.
Reader	View all compliance controls.

Request

URL. Use this URL to request a compliance control list, where <qualysapi.qualys.com> is the API server URL where your account is located. Replace the API server URL if your account is located on another platform.

```
https://<qualysapi.qualys.com>/api/2.0/fo/
compliance/control/
```

Method. The GET or POST access method may be used.

Authentication. Authentication is required for each API request. See Chapter 2, "Authentication Using the V2 APIs."

Parameters

The parameters used to make a control list request are described below.

Parameter	Description
action=list	(Required) Specifies the action type used to request a control list.
echo_request={**0** | 1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
details={**Basic** | All | None}	(Optional) Show the requested amount of information for each control. A valid value is: None — show control ID only Basic (default) — show control ID and basic control information: the control category, sub-category, statement, and technology information All — show control ID, basic control information, and framework mappings
ids={value}	(Optional) Show only certain control IDs and/or ID ranges. Multiple entries are comma separated. One or more control IDs/ranges may be specified. A control ID range entry is specified with a hyphen (for example, 3000-3250). Valid control IDs are required.
id_min={value}	(Optional) Show only controls which have a minimum control ID value. A valid control ID is required.
id_max={value}	(Optional) Show only controls which have a maximum control ID value. A valid control ID is required.

Parameter	Description
updated_after_datetime={value}	(Optional) Show only controls updated after a certain date/time. See "Date Filters" below.
created_after_datetime={value}	(Optional) Show only controls created after a certain date/time. See "Date Filters" below.
truncation_limit={value}	(Optional) The maximum number of control records processed per request. When not specified, the truncation limit is set to 1,000 host records. You may specify a value less than the default (1-999) or greater than the default (1001-1000000). If the requested list identifies more records than the truncation limit, then the XML output includes the <WARNING> element and the URL for making another request for the next batch of records. You can specify truncation_limit=0 for no truncation limit. This means that the output is not paginated and all the records are returned in a single output. WARNING: This can generate very large output and processing large XML files can consume a lot of resources on the client side. In this case it is recommended to use the pagination logic and parallel processing. The previous page can be processed while the next page is being downloaded.

Date Filters

The date/time is specified in YYYY-MM-DD{THH:MM:SSZ] format (UTC/GMT), like "2010-03-01" or "2010-03-01T23:12:00Z"

If you specify a date but no time as for example 2010-03-01, then the service automatically sets the time to 2010-03-01T00:00:00Z (the start of the day).

When date filters are specified using both input parameters for a single API request, both date filters are satisfied (ANDed).

XML Output

DTD for Control List

An API request for a technical control list returns XML output using the DTD which can be found at the following URL:

```
https://<qualysapi.qualys.com>/api/2.0/fo/
compliance/control/control_list_output.dtd
```

where <qualysapi.qualys.com> is the API server URL where your account is located.

The DTD for the compliance control list XML is provided in Appendix C.

Sample Control List Output

This sample control list output was produced for CID 1044 with **details=Basic**.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE CONTROL_LIST_OUTPUT SYSTEM
"https://qualyspapi.qualys.com/api/2.0/fo/compliance/control/contr
ol_list_output.dtd">

<CONTROL_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2010-03-16T22:53:05Z</DATETIME>
    <CONTROL_LIST>
      <CONTROL>
        <ID>1044</ID>
        <UPDATE_DATE>2010-02-12T00:00:00Z</UPDATE_DATE>
        <CREATED_DATE>2007-10-12T00:00:00Z</CREATED_DATE>
        <CATEGORY>Access Control Requirements</CATEGORY>
        <SUB_CATEGORY><![CDATA[Authorizations (Multi-user
ACL/role)]]></SUB_CATEGORY>
        <STATEMENT><![CDATA[Status of the
'O7_DICTIONARY_ACCESSIBILITY' setting in init.ora (ORACLE Data
Dictionary)]]></STATEMENT>
        <TECHNOLOGY_LIST>
          <TECHNOLOGY>
            <ID>7</ID>
            <NAME>Oracle 9i</NAME>
            <RATIONALE><![CDATA[The "O7_DICTIONARY_ACCESSIBILITY"
setting allows control/restrictions to be placed on the user's
SYSTEM privileges. If this parameter is set to TRUE, SYS schema
access will be allowed, which is the default for Oracle operations.
Restricting this system privilege with a setting of FALSE will
allow users or roles granted SELECT ANY TABLE access to objects in
the normal schema, but disallow access to objects in the SYS
schema, unless access is specifically granted.]]></RATIONALE>
          </TECHNOLOGY>
          <TECHNOLOGY>
            <ID>8</ID>
            <NAME>Oracle 10g</NAME>
```

```
            <RATIONALE><![CDATA[The "O7_DICTIONARY_ACCESSIBILITY"
setting allows control/restrictions to be placed on the user's
SYSTEM privileges. If this parameter is set to TRUE, SYS schema
access will be allowed, which is the default for Oracle operations.
Restricting this system privilege with a setting of FALSE will
allow users or roles granted SELECT ANY TABLE access to objects in
the normal schema, but disallow access to objects in the SYS
schema, unless access is specifically granted.]]></RATIONALE>
          </TECHNOLOGY>
          <TECHNOLOGY>
            <ID>9</ID>
            <NAME>Oracle 11g</NAME>
            <RATIONALE><![CDATA[The "O7_DICTIONARY_ACCESSIBILITY"
setting allows control/restrictions to be placed on the user's
SYSTEM privileges. If this parameter is set to TRUE, SYS schema
access will be allowed, which is the default for Oracle operations.
Restricting this system privilege with a setting of FALSE will
allow users or roles granted SELECT ANY TABLE access to objects in
the normal schema, but disallow access to objects in the SYS
schema, unless access is specifically granted.]]></RATIONALE>
          </TECHNOLOGY>
        </TECHNOLOGY_LIST>
      </CONTROL>
      <CONTROL>
        <ID>1045</ID>
        <UPDATE_DATE>2010-03-03T00:00:00Z</UPDATE_DATE>
        <CREATED_DATE>2007-10-12T00:00:00Z</CREATED_DATE>
        <CATEGORY>OS Security Settings</CATEGORY>
       <SUB_CATEGORY><![CDATA[System Settings (OSI layers 6-7)]]>
</SUB_CATEGORY>
        <STATEMENT><![CDATA[Status of the 'Clipbook' service
(Guidance = Disabled)]]></STATEMENT>
        <TECHNOLOGY_LIST>
          <TECHNOLOGY>
            <ID>1</ID>
            <NAME>Windows XP desktop</NAME>
            <RATIONALE><![CDATA[The 'Clipbook' service is used to
transfer Clipboard information across the LAN and is sent in clear
text.  The authentication required is a holdover from the 16-bit
'Network Dynamic Data Exchange' protocol, which is a 'network'
password among systems sharing the LAN, with a default set allow
READ for EVERYONE that has network access.  As this Windows service
is not required for any other system operations and increases
```

```
system vulnerability it should be disabled unless there is a
demonstrated need for its use set by the business.]]></RATIONALE>
          </TECHNOLOGY>
          <TECHNOLOGY>
            <ID>2</ID>
            <NAME>Windows 2003 Server</NAME>
            <RATIONALE><![CDATA[The 'Clipbook' service is used to
transfer Clipboard information across the LAN and is sent in clear
text.  The authentication required is a holdover from the 16-bit
'Network Dynamic Data Exchange' protocol, which is a 'network'
password among systems sharing the LAN, with a default set allow
READ for EVERYONE that has network access.  As this Windows service
is not required for any other system operations and increases
system vulnerability it should be disabled unless there is a
demonstrated need for its use set by the business.]]></RATIONALE>
          </TECHNOLOGY>
          <TECHNOLOGY>
            <ID>12</ID>
            <NAME>Windows 2000</NAME>
            <RATIONALE><![CDATA[The 'Clipbook' service is used to
transfer Clipboard information across the LAN and is sent in clear
text.  The authentication required is a holdover from the 16-bit
'Network Dynamic Data Exchange' protocol, which is a 'network'
password among systems sharing the LAN, with a default set allow
READ for EVERYONE that has network access.  As this Windows service
is not required for any other system operations and increases
system vulnerability it should be disabled unless there is a
demonstrated need for its use set by the business.]]></RATIONALE>
          </TECHNOLOGY>
</CONTROL_LIST_OUTPUT>
```

Compliance Policy List

The "Compliance Policy List" API v2 (the resource **`/api/2.0/fo/compliance/policy/`** with the parameter **`action=list`**) is used to view a list of compliance policies visible to the user. The compliance policy ID for each policy is listed in the policy list output. Policies in the XML output are sorted by compliance policy ID in ascending order. Optional input parameters support filtering the policy list output. Authentication is required for each API request. See Chapter 2, "Authentication Using the V2 APIs."

By default, the compliance policy list output shows all compliance policies in the subscription. Optional input parameters allow you to set filters to restrict the policy list output to policies with certain compliance policy IDs.

The user has the ability to select the amount of additional information to include for each policy in the output. By default, this basic policy information is included: the policy ID, the policy title, the date/time when the policy was created and last modified, a list of asset group IDs identifying the asset groups assigned to the policy, a list of controls included in the policy, and a glossary of compliance policy data in the policy list output.

Use the **`details`** input parameter to select another level of detail to be included in the policy list output. Two parameter settings are available. Specify **`details=All`** to show the basic compliance policy information, and a list of framework mappings, and a glossary of compliance policy data in the policy list output. Specify **`details=None`** to list the compliance policy ID only.

The optional glossary in the compliance policy list output includes:

Output	Description
Asset Group List	A list of asset groups assigned to policies in the policy list output.
IP List	A list of IP addresses/ranges in asset groups assigned to policies in the policy list output.
User List	A list of users who created one or more policies or edited one or more policies in the policy list output. For an edited policy, the last user who modified the policy is listed.

Maximum Policies per API Request

A maximum of 1,000 compliance policy records can be processed per request. If the requested list identifies more than 1,000 policies, then the XML output includes the <WARNING> element and instructions for making another request for the next batch of policy records.

User Permissions

User permissions are described below.

User Role	Permissions
Manager	View all compliance policies in subscription. View asset group information for all asset groups assigned to policies.
Auditor	View all compliance policies in subscription. View asset group information for all asset groups assigned to policies.
Unit Manager	View all compliance policies in subscription. View asset group information for asset groups assigned to policies, when the user has permission to view these asset groups. This user can view groups assigned to the user's business unit, and groups created by any user in the same business unit.
Scanner	View all compliance policies in subscription. View asset group information for asset groups assigned to policies, when the user has permission to view these asset groups. This user can view groups assigned to the user account, and groups created by the user.
Reader	View all compliance policies in subscription. View asset group information for asset groups assigned to policies, when the user has permission to view these asset groups. This user can view groups assigned to the user account, and groups created by the user.

User Permissions — Asset Group Information

Asset group information included in the policy list output includes the following, as defined for each asset group: asset group ID, title, and assigned IP addresses. Users are granted permission to view asset group information assigned to policies when the user has permission to view the asset groups.

For example, when a user makes a request for a compliance policy list and the user does not have permission to view asset groups that are assigned to the target policies, then the asset group information does not appear in the policy list output. The asset group IDs are not listed under the <POLICY> section, and the asset group title and assigned IP addresses are not listed under the <GLOSSARY> section.

In a case where a user makes a request for a compliance policy list and the user does not have permission to see one or more asset groups assigned to a target policy, the following information is provided in the compliance policy list output:

<POLICY> section. The attribute "has_hidden_data=1" is returned in the <POLICY> section in the <ASSET_GROUP_IDS> element. This indicates that the user does not have permission to see one or more asset groups in the policy. When this attribute is present, only the asset group IDs that the user has permission to see, if any, are listed in the <ASSET_GROUP_IDS> element.

<GLOSSARY> section. Asset group information is not displayed for asset groups assigned to compliance policies that the user does not have permission to see.

<WARNING_LIST> section. A warning message is returned for informational purposes. This indicates that at least one of the compliance policies in the output has one or more asset groups that the user does not have permission to see.

Request

URL. Use this URL to request a compliance policy list, where `<qualysapi.qualys.com>` is the API server URL where your account is located. Replace the API server URL if your account is located on another platform.

```
https://<qualysapi.qualys.com>/api/2.0/fo/
compliance/policy/
```

Method. The GET or POST access method may be used.

Authentication. Authentication is required for each API request. See Chapter 2, "Authentication Using the V2 APIs."

Parameters

The parameters used to make a compliance policy list request are described below.

Parameter	Description
action=list	(Required) Specifies the action type used to request a policy list.
echo_request={**0**|1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.

Parameter	Description
details={**Basic** | All | None}	(Optional) Show requested amount of information for each policy. A valid value is: None — show policy ID only Basic (default) — show policy ID, basic policy information including: asset group ID, asset group title, compliance control list, and glossary of compliance policy data in the output All — show policy ID, basic policy information, and glossary of compliance policy data in the output
ids={value}	(Optional) Show only certain policy IDs and/or ID ranges. One or more policy IDs/ranges may be specified. Multiple entries are comma separated. A policy ID range entry is specified with a hyphen (for example, 160-165). Valid policy IDs are required.
id_min={value}	(Optional) Show only policies which have a minimum policy ID value. A valid policy ID is required.
id_max={value}	(Optional) Show only policies which have a maximum policy ID value. A valid policy ID is required.

XML Output

DTD for Compliance Policy List

An API request for a compliance policy list returns XML output using the DTD which can be found at the following URL:

```
https://<qualysapi.qualys.com>/api/2.0/fo/
compliance/policy/policy_list_output.dtd
```

where <qualysapi.qualys.com> is the API server URL where your account is located.

The DTD for the compliance policy list XML output is provided in Appendix C.

Sample Compliance Policy List

Sample compliance policy list output (fragment) with **details=Basic** is below.

```
<!DOCTYPE POLICY_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/
policy_list_output.dtd">
```

```
<POLICY_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2008-05-21T13:57:17Z</DATETIME>
    <POLICY_LIST>
      <POLICY>
        <ID>165</ID>
        <TITLE>Small Policy</TITLE>
        <CREATED>
          <DATETIME>2007-10-23T00:44:10Z</DATETIME>
          <BY>acme_bb</BY>
        </CREATED>
        <LAST_MODIFIED>
          <DATETIME>2008-07-17T18:54:53Z</DATETIME>
          <BY>acme_bb</BY>
        </LAST_MODIFIED>
        <ASSET_GROUP_IDS>14392,14962</ASSET_GROUP_IDS>
        <CONTROL_LIST>
          <CONTROL>
            <ID>1044</ID>
            <STATEMENT><![CDATA[Access to the data dictionary
shall be restricted]]></STATEMENT>
          </CONTROL>
          <CONTROL>
            <ID>1045</ID>
            <STATEMENT><![CDATA[The Clipbook service shall be
disabled]]></STATEMENT>
          </CONTROL>
          ...
          <CONTROL>
            <ID>1131</ID>
            <STATEMENT><![CDATA[The TFTP daemon shall be
disabled]]></STATEMENT>
          </CONTROL>
        </CONTROL_LIST>
      </POLICY>
      <POLICY>
        <ID>231</ID>
        <TITLE>My Policy for Windows</TITLE>
        <CREATED>
          <DATETIME>2007-11-01T23:05:15Z</DATETIME>
          <BY>acme_tt1</BY>
        </CREATED>
```

```
        <LAST_MODIFIED>
          <DATETIME>2008-04-11T22:29:20Z</DATETIME>
          <BY>acme_bb</BY>
        </LAST_MODIFIED>
      </POLICY>
    ...
    </POLICY_LIST>
    <GLOSSARY>
      <ASSET_GROUP_LIST>
        <ASSET_GROUP>
          <ID>14392</ID>
          <TITLE><![CDATA[My compliance group]]></TITLE>
        </ASSET_GROUP>
        <ASSET_GROUP>
          <ID>14962</ID>
          <TITLE><![CDATA[1]]></TITLE>
        </ASSET_GROUP>
      </ASSET_GROUP_LIST>
    </GLOSSARY>
  </RESPONSE>
</POLICY_LIST_OUTPUT>
```

Compliance Policy - Import and Export

The "Import/Export Compliance Policy" API v2 (the resource **`/api/2.0/fo/compliance/policy/`** with the parameter **`action=import|export`**) gives you the ability to import and export compliance policies.

Your account must have the Policy Compliance application enabled in order to import and export policies using this API.

Export a compliance policy

You can export a compliance policy, that exists in your account, to an XML file.

Permissions: If you're not a Manager this permission must be turned on in your account: Manage PC module

Supported HTTP methods: GET and POST

Input Parameters:

Parameter	Description
action=export	(Required) Specifies the action type for exporting the policy.
echo_request={**0** | 1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
id={value} or title={value}	(Required) The ID or the title of the policy you want to export.

Sample Request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST"
-d "action=export&id=853744"
"https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/"
```

XML output:

```
<?xml version="1.0 encoding=UTF-8" ?>
<DOCTYPE POLICY_EXPORT_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2/fo/compliance/policy/policy_ex
port_output.dtd">
<POLICY>
    <TITLE><![CDATA[My Policy]]></TITLE>
    <EXPORTED><![CDATA[2013-07-17T18:19:57Z]]></EXPORTED>
```

```
    <COVER_PAGE><![CDATA[]]></COVER_PAGE>
    <TECHNOLOGIES total="1">
        <TECHNOLOGY>
            <ID>1</ID>
            <NAME>Windows XP desktop</NAME>
        </TECHNOLOGY>
    </TECHNOLOGIES>
    <SECTIONS total="1">
        <SECTION>
            <NUMBER>1</NUMBER>
            <HEADING><![CDATA[Default section]]></HEADING>
            <CONTROLS total="20">
                <CONTROL>
                    <ID>1111</ID>
                    <TECHNOLOGIES total="1">
                        <TECHNOLOGY>
                            <ID>1</ID>
                            <NAME>Windows XP desktop</NAME>
                            <EVALUATE
checksum="74378d12a39f82721a3cb156dee58c663a650a9ce422bd311b5e5443
c2a20f14">&lt;CTRL&gt;&lt;NOT&gt;&lt;DP&gt;&lt;K&gt;auth.general.l
ogintext&lt;/K&gt;&lt;OP&gt;re&lt;/OP&gt;&lt;V&gt;&lt;![CDATA[^(\s
*|314159265358979|161803399999999)$]]&gt;&lt;/V&gt;&lt;/DP&gt;&lt;
/NOT&gt;&lt;/CTRL&gt;</EVALUATE>
                        </TECHNOLOGY>
                    </TECHNOLOGIES>
                </CONTROL>
        </SECTION>
    </SECTIONS>
</POLICY>
```

Policy Export Output DTD:

An API request returns output using the policy export output DTD which can be found at the following URL (where qualysapi.qualys.com is the API server where your account is located):

```
https://qualysapi.qualys.com/api/2/fo/compliance/policy/policy_
export_output.dtd
```

This DTD is described in Appendix C.

Import a compliance policy

You can import a compliance policy, defined in an XML file, into your account.

Permissions: If you're not a Manager this permission must be turned on in your account: Manage PC module

Supported HTTP Method: POST

Input Parameters:

Parameter	Description
action=import	(Required) Specifies the action type for importing the policy.
echo_request={**0**|1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
xml_file	(Required) The file containing the policy details.
title={value}	(Required) The title of the new policy.

Sample Request:

```
curl -H "X-Requested-With: Curl Sample" -H "Content-type:
text/xml" --data-binary @policy.xml -u "USERNAME:PASSWORD"
"https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/?action
=import&title=My+Policy"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
   <RESPONSE>
      <DATETIME>2013-01-15T21:32:40Z</DATETIME>
      <TEXT>Successfully imported compliance policy</TEXT>
      <ITEM_LIST>
         <ITEM>
         <KEY>ID</KEY>
         <VALUE>136992</VALUE>
      </ITEM>
      <ITEM>
         <KEY>TITLE</KEY>
         <VALUE>My Policy</VALUE>
```

```
      </ITEM>
   </ITEM_LIST>
</RESPONSE>
</SIMPLE_RETURN>
```

Compliance Policy - Merge

The "Compliance Policy Merge" API v2 (the resource **`/api/2.0/fo/compliance/policy/`** with the parameter **`action=merge`**) allows you to merge (combine) 2 or more compliance policies using Qualys Policy Compliance (PC). You can choose to merge some or all parts of a new policy into an existing one. Also you can preview merge changes before saving them. This API is available to Managers and Auditors.

For example, say you imported a policy from our library (Policy A) and configured it to add asset groups, controls and sections. Later we might release an updated version of this policy (Policy B) with new controls and technologies. In this scenario you can use the Policy Merge API to add the new controls and technologies from Policy B into Policy A (your existing policy) without losing the asset groups, controls and sections you added.

Parameters

The policy merge input parameters give you flexibility with merging different parts of a new policy (Policy B) into an existing one (Policy A). For example you can choose to update controls with newer definitions, replace asset groups, and add new technologies and controls. By default no changes are applied to your existing policy unless parameters are specified (see below).

Parameter	Description
action=merge	(Required)
id={value}	(Required) The ID of the policy that will be updated with merged content (let's call this Policy A).
merge_policy_id={value} -or- policy XML data	(Required) Tell us the policy with the content that will be merged into Policy A (let's call this Policy B). You can specify a policy ID using "merge_policy_id" or policy XML data. To upload XML data, use this syntax: --data-binary @path_to_xml_file.xml
	These options are mutually exclusive: policy XML data and replace_asset_groups.
replace_cover_page={**0**|1}	(Optional) Set replace_cover_page=1 to replace the cover page in Policy A with the cover page in Policy B.
replace_asset_groups={**0**|1}	(Optional) Set replace_asset_groups=1 to replace asset groups in Policy A with asset groups in Policy B.
	These options are mutually exclusive: add_asset_groups and replace_asset_groups.
add_asset_groups={**0**|1}	(Optional) Set add_asset_groups=1 to add new asset groups, i.e. add asset groups from Policy B if they are not already present in Policy A.

Parameter	Description
add_new_technologies={**0**|1}	(Optional) Set add_new_technologies=1 to add new technologies, i.e. add technologies from Policy B if they are not already in Policy A.
add_new_controls={**0**|1}	(Optional) Set add_new_controls=1 to add new controls, i.e. add controls from Policy B if they are not already in Policy A.
update_section_heading={**0**|1}	(Optional) Set update_section_heading=1 to replace the section heading in Policy A with the one in Policy B, based on section number (applies only to common sections). This parameter must be specified with: **add_new_controls** or **update_existing_controls**.
update_existing_controls={**0**|1}	(Optional) Set update_existing_controls=1 to replace the common controls in Policy A with the ones in Policy B. These are controls that exist in both policies. (Controls will not be removed).
preview_merge={**0**|1}	(Optional) Set preview_merge= 1 to view the changes merged into Policy A without saving them.

XML Output

DTD for Policy Merge Results Output

The DTD for policy merge results output XML can be found at this URL:

```
https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/policy_merge_result_output.dtd"
```

If your account is not located on Qualys Platform 1, please replace "qualysapi.qualys.com" with your platform URL.

Policy Merge Request 1 - preview merged policy

Policy ID 15993 (Policy A) will be updated with content merged from policy ID 15994 (Policy B) and the XML output will show the merged policy in preview mode. Policy changes will not be saved in Policy 15993 since the request includes "preview_merge=1".

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl"
"https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/?
action=merge&id=15993&merge_policy_id=15994&replace_cover_page=1&a
dd_new_asset_groups=1&add_new_technologies=1&update_section_headin
g=1&add_new_controls=1&update_existing_controls=1&preview_merge=1"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE POLICY_MERGE_RESULT_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/policy_
merge_result_output.dtd">
<POLICY_MERGE_RESULT_OUTPUT>
  <RESPONSE>
    <DATETIME>2013-12-24T05:28:04Z</DATETIME>
    <POLICY_MERGE_RESULT>
      <NOTE>Policy changes were not merged or saved since the
request had preview_merge=1.</NOTE>
      <NEW_COVER_PAGE><![CDATA[My Cover Page]]></NEW_COVER_PAGE>
      <ASSET_GROUPS_ADDED>
        <ASSET_GROUP>
          <ID>424422</ID>
          <NAME><![CDATA[<script>alert("xss");</script>]]></NAME>
        </ASSET_GROUP>
        <ASSET_GROUP>
          <ID>424577</ID>
          <NAME><![CDATA[10.10.32.26]]></NAME>
        </ASSET_GROUP>
      </ASSET_GROUPS_ADDED>
      <TECHNOLOGIES_ADDED>
        <TECHNOLOGY>
          <ID>1</ID>
          <NAME>Windows XP desktop</NAME>
        </TECHNOLOGY>
      </TECHNOLOGIES_ADDED>
      <SECTIONS_UPDATED>
        <SECTION>
          <ID>1</ID>
          <HEADING><![CDATA[First section]]></HEADING>
        </SECTION>
        <SECTION>
          <ID>2</ID>
          <HEADING><![CDATA[Second section]]></HEADING>
        </SECTION>
      </SECTIONS_UPDATED>
      <SECTIONS>
```

```
        <SECTION>
          <ID>1</ID>
          <CONTROLS_UPDATED>
            <CONTROL>
              <ID>1061</ID>
            </CONTROL>
          </CONTROLS_UPDATED>
        </SECTION>
        <SECTION>
          <ID>2</ID>
          <CONTROLS_ADDED>
            <CONTROL>
              <ID>1045</ID>
            </CONTROL>
            <CONTROL>
              <ID>1048</ID>
            </CONTROL>
          </CONTROLS_ADDED>
        </SECTION>
      </SECTIONS>
    </POLICY_MERGE_RESULT>
  </RESPONSE>
</POLICY_MERGE_RESULT_OUTPUT>
```

Policy Merge Request 2 - save merged policy

Policy ID 15993 (Policy A) will be updated with content merged from policy ID 15994 (Policy B). The merged policy will be saved in policy 15993.

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl"
"https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/?
action=merge&id=15993&merge_policy_id=15994&replace_cover_page=1&a
dd_new_asset_groups=1&add_new_technologies=1&update_section_headin
g=1&add_new_controls=1&update_existing_controls=1&preview_merge=0"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE POLICY_MERGE_RESULT_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/policy_
merge_result_output.dtd">
```

```
<POLICY_MERGE_RESULT_OUTPUT>
  <RESPONSE>
    <DATETIME>2013-12-24T05:31:26Z</DATETIME>
    <POLICY_MERGE_RESULT>
      <NOTE>Policy changes have been merged successfully.</NOTE>
      <NEW_COVER_PAGE><![CDATA[My Cover Page]]></NEW_COVER_PAGE>
      <ASSET_GROUPS_ADDED>
        <ASSET_GROUP>
          <ID>424422</ID>
...
    </POLICY_MERGE_RESULT>
  </RESPONSE>
</POLICY_MERGE_RESULT_OUTPUT>
```

Policy Merge Request 3 - pass policy XML, preview merged policy

Policy ID 15993 (Policy A) will be updated with content merged from the policy defined in the file "path_to_policy_xml_file.xml." The merged changes will not be saved in policy 15993 since the request includes "preview_merge=1".

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -H
"Content-type: text/xml"
"https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/?
action=merge&id=15993&replace_cover_page=1&replace_asset_groups=1&
add_new_technologies=1&update_section_heading=1&add_new_controls=1
&update_existing_controls=1&preview_merge=1" --data-binary
@/home/aamin/PC_XML/path_to_policy_xml_file.xml>
```

XMLoutput:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE POLICY_MERGE_RESULT_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/policy_
merge_result_output.dtd">
<POLICY_MERGE_RESULT_OUTPUT>
  <RESPONSE>
    <DATETIME>2013-12-24T05:38:26Z</DATETIME>
    <POLICY_MERGE_RESULT>
      <NOTE>Policy changes were not merged or saved since the
request had preview_merge=1.</NOTE>
      <NEW_COVER_PAGE><![CDATA[My Cover Page]]></NEW_COVER_PAGE>
```

```
      <SECTIONS_UPDATED>
        <SECTION>
          <ID>1</ID>
          <HEADING><![CDATA[First section]]></HEADING>
        </SECTION>
        <SECTION>
          <ID>2</ID>
          <HEADING><![CDATA[Second section]]></HEADING>
        </SECTION>
      </SECTIONS_UPDATED>
      <SECTIONS>
        <SECTION>
          <ID>1</ID>
          <CONTROLS_UPDATED>
            <CONTROL>
              <ID>1061</ID>
            </CONTROL>
          </CONTROLS_UPDATED>
        </SECTION>
        <SECTION>
          <ID>2</ID>
          <CONTROLS_ADDED>
            <CONTROL>
              <ID>1045</ID>
            </CONTROL>
            <CONTROL>
              <ID>1048</ID>
            </CONTROL>
          </CONTROLS_ADDED>
        </SECTION>
      </SECTIONS>
    </POLICY_MERGE_RESULT>
  </RESPONSE>
</POLICY_MERGE_RESULT_OUTPUT>
```

Compliance Policy - Manage Asset Groups

Manage asset groups for your compliance policies using the "Compliance Policy" API v2 (/api/2.0/fo/compliance/policy). Parameters allow you to add, remove and set asset groups for a policy. You must have permission to modify the policy you want to update.

action=add_asset_group_ids

Use this action to add asset groups to a specified policy. Parameters:

Parameter	Description
id={value}	Policy ID for the policy you want to update.
asset_group_ids={value}	Asset groups IDs for the asset groups you want to add to the policy specified in "id". Multiple IDs are comma separated. Each asset group must have at least 1 assigned IP address.

API request:

```
 curl -H "X-Requested-With: curl" -u "USERNAME:PASSWD" -X POST -d
"id=43400&asset_group_ids=649737,649736"
"https://qualysapi.qualys.com//api/2.0/fo/compliance/policy/?actio
n=add_asset_group_ids"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2014-09-11T09:06:17Z</DATETIME>
    <TEXT>Compliance Policy successfully modified.</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>43400</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

action=remove_asset_group_ids

Use this action to remove asset groups from a specified policy. Parameters:

Parameter	Description
id={value}	Policy ID for the policy you want to update.
asset_group_ids={value}	Asset groups IDs for the asset groups you want to delete from the policy specified in "id". Multiple IDs are comma separated.

API request:

```
curl -H "X-Requested-With: curl" -u "USERNAME:PASSWD" -X POST -d
"id=43400&asset_group_ids=649737,649736"
"https://qualysapi.qualys.com//api/2.0/fo/compliance/policy/?actio
n=remove_asset_group_ids"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2014-09-11T09:06:17Z</DATETIME>
    <TEXT>Compliance Policy successfully modified.</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>43400</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

action=set_asset_group_ids

Use this action to reset the asset groups for a specified policy. Any assigned asset groups not specified in this request will be removed. Parameters:

Parameter	Description
id={value}	Policy ID for the policy you want to update.
asset_group_ids={value}	Asset groups IDs for the asset groups you want to assign to the policy specified in "id". Multiple IDs are comma separated. Each asset group must have at least 1 assigned IP address.

API request:

```
curl -H "X-Requested-With: curl" -u "USERNAME:PASSWD" -X POST -d
"id=43400&asset_group_ids=649737,649736"
"https://qualysapi.qualys.com//api/2.0/fo/compliance/policy/?
action=set_asset_group_ids"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2014-09-11T09:07:43Z</DATETIME>
    <TEXT>Compliance Policy successfully modified.</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>ID</KEY>
        <VALUE>43400</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

Compliance Posture Information

The "Compliance Posture Info" API v2 (the resource **`/api/2.0/fo/compliance/posture/info/`** with the parameter **`action=list`**) is used to view current compliance posture data (info records) for hosts within the user's account. Each compliance posture info record includes a compliance posture ID and other attributes. Optional input parameters support filtering the posture info record output. Authentication is required for each API request. See Chapter 2, "Authentication Using the V2 APIs."

Each compliance posture info record in the output includes:

Output	Description
Compliance Posture ID	The service assigns a unique value to each compliance posture info record.
Host ID	Identifies a host.
Control ID	Identifies a technical control.
Technology ID	Identifies a technology.
Instance	Identifies a technology instance, when applicable.
Compliance Status	Passed, Failed or Error. An error, only assigned to a custom control, indicates control evaluation failed (and the ignore errors configuration option for the control was not selected).
Exception	Identifies an exception assignee and status, if an exception has been created.

The user has the ability to select the amount of information to include in the posture information output. By default, basic posture information is included: the posture ID, host ID, control ID, technology ID, technology instance (when applicable), and the compliance status. If an exception has been created, this full exception information is also included: the exception assignee and status, the date/time when the exception was created, when it was last modified, the user who took these actions on the exception, and the date when the exception is set to expire. A glossary of compliance posture information identifies: basic host information and basic control information.

Use the **`details`** input parameter to select another level of detail to be included in the policy information output.

By default, the posture information output shows posture information for all hosts (IP addresses) in asset groups assigned to the selected policy, provided the user has permission to view the hosts themselves. If you have a sub-account like a Unit Manager, Scanner or Reader, the posture information output only includes hosts that the your account has permission to see. Optional input parameters allow you to set filters to

restrict the posture information output to postures info records with certain IP addresses, host IDs, compliance control IDs, compliance posture IDs, posture info records with changes in status since a specified date, and posture info records with a certain compliance status (Passed, Failed or Error).

The optional glossary in the compliance posture information output includes:

Output	Description
User List	List of users who created, modified, or added comments to exceptions in compliance posture info records which are included in the posture information output. For a policy that was edited, the user who most recently edited the exception is listed.
Host List	List of hosts in compliance posture info records which are included in the posture information output. This basic host information is included: host ID, IP address, and tracking method. When **`details=All`** is specified, this additional information is included: last vulnerability scan date/time, last compliance scan date/time.
Control List	List of controls in compliance posture info records which are included in the posture information output. When **`details=All`** is specified, this additional information is included: rationale information and technology information for each control.
Technology List	List of technologies for controls in compliance posture info records which are included in the posture list output. This information is included only when **`details=All`** is specified.
Evidence List	List of evidence information for control data points.

Maximum Postures per API Request

The output of the Compliance Posture Info API is paginated when your API request identifies a single policy to report on using the “policy_id” input parameter. In this case, a maximum of 5,000 posture info records are returned per request by default. You can customize the page size (i.e. the number of posture info records) by using the parameter “truncation_limit=10000” for instance if you want to return pages with 10,000 records.

User Permissions

All users have permission view posture information for hosts (IP addresses) in asset groups assigned to the selected policy, when the hosts are available to the user based on user account settings. User permissions are described below.

User Role	Permissions
Manager	View compliance postures for all hosts (IP addresses) in asset groups assigned to the selected policy.
Auditor	View compliance postures for all hosts (IP addresses) in asset groups assigned to the selected policy.
Unit Manager	View compliance postures for all hosts (IP addresses) in asset groups assigned to the selected policy, when the hosts are included in the user's business unit.
Scanner	View compliance postures for all hosts (IP addresses) in asset groups assigned to the selected policy, when the hosts are included in the user's account.
Reader	View compliance postures for all hosts (IP addresses) in asset groups assigned to the selected policy, when the hosts are included in the user's account.

User Permissions: Asset Group IPs

All users have permission to view posture information for all hosts (IP addresses) in the asset groups assigned to the selected policy provided they have permission to view the hosts themselves. This permission is granted even when users do not have permission to view the asset groups assigned to the policy.

For example, when a user makes a request for compliance posture information for "Policy A" and this policy has one assigned asset group "Hong Kong", and the user does not have permission to view this asset group, then the user does have permission to view compliance posture info records for all the IP addresses in the asset group "Hong Kong" provided the IP addresses in the group "Hong Kong" are visible to the user.

Request

URL. Use this URL to request a compliance posture information, where `<qualysapi.qualys.com>` is the API server URL where your account is located. Replace the API server URL if your account is located on another platform.

```
https://<qualysapi.qualys.com>/api/2.0/fo/
compliance/posture/info/
```

Method. The GET or POST access method may be used.

Authentication. Authentication is required for each API request. See Chapter 2, "Authentication Using the V2 APIs."

Parameters

The parameters used to make a compliance info records request are described below.

Parameter	Description
action=list	(Required) Specifies the action type used to request compliance policy info records.
policy_id={value}	(**policy_id** or **policy_ids** is required) Show compliance posture info records for a specified policy. A valid policy ID is required.
	The parameters **policy_id** and **policy_ids** cannot be specified in the same request.
policy_ids={value}	(**policy_id** or **policy_ids** is required) Show compliance posture info records for multiple policies - up to 10 policies may be requested. Provide a comma-separated list of valid policy IDs. When this parameter is specified, all posture data is downloaded (and the "truncation_limit" parameter is invalid).
	The parameters **policy_id** and **policy_ids** cannot be specified in the same request. When **policy_ids** is specified, **truncation_limit** is invalid. For CSV output, **policy_id** must be specified (and **policy_ids** is invalid).
echo_request={**0** | 1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
output_format={value}	(Optional) The output format. A valid value is: xml (default), csv (posture data and metadata i.e. summary and warning data), csv_no_metadata (posture data only, no metadata). For CSV output you can include only one policy for this reason **policy_id** is required.
details={**Basic** | All | None | Light}	(Optional) Show a certain amount of information for each compliance posture info record. A valid value is: None — show posture info and minimum exception information (assignee and status) if appropriate Basic (default) — show posture info, full exception information if appropriate, and a minimum glossary (basic info for hosts and controls) Light — show posture info, exception info if appropriate, and a limited glossary (host info and last scan date/time, control ID, and evidence info All — show posture info (including the percentage of controls that passed for each host), exception info if appropriate, posture summary (the number of assets, controls, and control instances evaluated) and a glossary (host info and last scan date/time), control info, technology info, evidence info

Parameter	Description
ips={value}	(Optional) Show only compliance posture info records for compliance hosts which have certain IP addresses/ranges. One or more IP addresses/ranges may be specified. Multiple IPs/ranges are comma separated.
host_ids={value}	(Optional) Show only compliance posture info records for compliance hosts which have certain host IDs and/or ID ranges. One or more host IDs/ranges may be specified. Multiple entries are comma separated. A host ID range entry is specified with a hyphen (for example, 123-125). Valid host IDs are required.
control_ids={value}	(Optional) Show only compliance posture info records for controls which have certain control IDs and/or ranges. One or more control IDs/ranges may be specified. Multiple entries are comma separated. An control ID range entry is specified with a hyphen (for example, 1200-1300). Valid control IDs are required.
ids={value}	(Optional) Show only compliance posture info records for certain compliance posture IDs and/or ID ranges. One or more posture IDs/ranges may be specified. Multiple entries are comma separated. A posture ID range entry is specified with a hyphen (for example, 1-10). Valid posture IDs are required.
id_min={value}	(Optional) Show only compliance posture info records which have a minimum ID value. A valid posture ID is required.
id_max={value}	(Optional) Show only compliance posture info records which have a maximum ID value. A valid posture ID is required.
status_changes_since={date}	(Optional) Show compliance posture info records when the compliance status was changed since a certain date and time (optional). If the policy itself was changed, a warning message is generated. The date/time is specified in YYYY-MM-DD[THH:MM:SSZ] format (UTC/GMT), like “2008-05-01” or “2008-05-01T23:12:00Z”.
asset_group_ids={value}	(Optional) Show only hosts in certain asset groups. Provide a comma-separated list of asset group IDs for the asset groups you want to download compliance posture data for. The asset groups specified do not need to be assigned to the one or more policies requested. Posture data will be returned as long as there are common hosts specified by “asset_group_ids” and asset groups that are assigned to the policies requested.
status={Passed | Failed | Error}	(Optional) Show only compliance posture info records which have a posture status of Passed, Failed or Error. By default, records with the status Passed, Failed and Error are listed.

Parameter	Description
criticality_labels={value}	(Optional) Show only compliance posture info records for controls which have certain criticality labels. One or more criticality labels (e.g. SERIOUS, CRITICAL, URGENT) may be specified. Multiple entries are comma separated.
	The parameters **criticality_labels** and **criticality_values** cannot be specified in the same request.
criticality_values={value}	(Optional) Show only compliance posture info records for controls which have certain criticality values. One or more criticality values (0-5) may be specified. Multiple entries are comma separated.
	The parameters **criticality_labels** and **criticality_values** cannot be specified in the same request.
truncation_limit={value}	(Optional) The parameter is valid only when the API request is for a single policy and the **policy_id** parameter is specified. By default, a limit of 5,000 posture info records are returned per request (when "policy_id" is specified). You may specify a value less than the default (1-4999) or greater than the default (5001-1000000) to configure the number records returned per request. If the requested list identifies more records than the truncation limit, then the XML output includes the <WARNING> element and the URL for making another request for the next batch of records. You can specify truncation_limit=0 for no truncation limit. This means that the output is not paginated and all the records are returned in a single output. WARNING: This can generate very large output and processing large XML files can consume a lot of resources on the client side. In this case it is recommended to use the pagination logic and parallel processing. The previous page can be processed while the next page is being downloaded.

XML Output

DTD for Compliance Posture Information

An API request for compliance posture information returns XML output using the DTD which can be found at the following URL:

```
https://<qualysapi.qualys.com>/api/2.0/fo/
compliance/posture/info/posture_info_list_output.dtd
```

where <qualysapi.qualys.com> is the API server URL where your account is located.

The DTD for the compliance posture information XML output is provided in Appendix C.

Control Criticality

Control Criticality is a feature in Policy Compliance that provides ratings for controls, including the ability to customize ratings at the control level and at the policy level. Several APIs include control criticality in the API output.

Control Criticality must be enabled in your account — By default, control criticality will not be enabled while we are updating the default criticality settings in the control library. If you want this feature, please contact Support or your Technical Account Manager.

SCAP Cyberscope Report

Under the Federal Information Security Management Act of 2002 (FISMA), government agencies are obliged to report on their information security statuses using a common tool called Cyberscope. Qualys customers with the SCAP module enabled can scan their network and generate Cyberscope compatible XML reports, using new API functions, to meet these requirements.

Qualys provides 3 different API functions for generating Cyberscope compatible XML reports as described below. The Cyberscope reports generated using these API functions return XML output in LASR format. For information on the Cyberscope report specification and the LASR format please see:
http://scap.nist.gov/use-case/cyberscope

SCAP Scan Results API

The “SCAP Scan Results” API v2 (the resource **/api/2.0/fo/asset/host/cyberscope/fdcc/scan/**) creates a Cyberscope report using scan results for a particular SCAP scan in the user’s account. An SCAP scan ID or scan reference is required as input. The service uses only the data in the raw scan results to generate the report. When the parameters organisation_name1, organisation_name2, and organisation_name3 are specified, the <ai:Organization> elements are included in the XML report.

Permissions: Users have permission to run this API function when the SCAP module is enabled for the user's subscription. Sub-accounts (Unit Managers, Scanners and Readers) must have the "Manage compliance" permission.

Sample 1. Select SCAP Scan by Scan ID

Use the **scan_id** parameter to select an SCAP scan by scan ID. (A scan ID or reference number is required.)

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl"
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/cyberscope/fdc
c/scan/?scan_id=4244823&organisation_name1=Name1&organisation_name2=Na
me2&organisation_name3=Name3"
```

To obtain the SCAP scan ID, log into the Qualys application and go to PC/SCAP > Scans > SCAP Scans to view the SCAP scans in your account. Hover over the SCAP scan that you’re interested in and view the scan results (select View from the Quck Actions menu). You’ll see the scan results URL in your browser and the scan ID value appears in the "id" parameter, as shown in this sample URL:

```
https://qualyguard.qualys.com/fo/report/fdcc/fdcc_scan_result.php?
id=4297720
```

Sample 2. Select SCAP Scan by Scan Reference

Use the **scan_ref** parameter to select an SCAP scan by scan reference number. (A scan reference number or scan ID is required.)

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl"
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/cyberscope/fdc
c/scan/?scan_ref=qscap/1337984725.4360&organisation_name1=Name1&organ
isation_name2=Name2&organisation_name3=Name3"
```

Sample 3. IPs Filter

Use the optional **ips** parameter to include only certain IP addresses in the report. You can enter a single IP, multiple IPs and/or IP ranges. Multiple entries are comma separated.

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl"
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/cyberscope/fdc
c/scan/?scan_id=4268027&ips=10.10.26.183&organisation_name1=Name1&or
ganisation_name2=Name2&organisation_name3=Name3"
```

SCAP Policy Results API

The "SCAP Policy Results" API v2 (the resource **/api/2.0/fo/asset/host/cyberscope/fdcc/policy/**) creates a Cyberscope report using scan results data saved for a particular SCAP policy in the user's account. A policy ID is required as input. These parameters allow users to customize the required "OrganisationName" elements in the XML report: organisation_name1, organisation_name2, and organisation_name3.

The service uses automatic SCAP policy data for a selected policy and reports this in the datapoint <sr:DataPoint id:"configuration_management_agency_deviations">. The services uses the evidence data for the special rule "security_patches_up_to_date" and reports this in the datapoint <sr:DataPoint id:"vulnerability_management_product_vulnerabilities">.

Permissions: Users have permission to run this API function when the SCAP module is enabled for the user's subscription and sub-accounts (Unit Managers, Scanners and Readers) have the "Manage compliance" permission.

Sample 1. Select an SCAP Policy

Use the **policy_id** parameter to select an SCAP policy. Hosts in the policy will be included in the report unless filters are specified using the parameter **ips** and/or **as_ids**.

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl"
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/cyberscope/fdc
c/policy/?policy_id=30231&organisation_name1=Name1&organisation_name2=
Name2&organisation_name3=Name3"
```

To obtain the SCAP policy ID, log into the Qualys application and go to PC/SCAP > Policies to view the policies in your account. Hover over the SCAP policy that you're interested in and edit it (select Edit from the Quck Actions menu). You'll see the policy editor URL in your browser and the policy ID value appears in the "id" parameter, as shown in this sample URL:

```
https://qualyguard.qualys.com/fo/fdcc/edit_policy.php?id=12345&re
fresh_parent=1
```

Sample 2. IPs Filter

Use the **ips** parameter to include only hosts with the specified IP addresses. Enter a single IP, multiple IPs and/or IP ranges using the **ips** parameter. Multiple entries are comma separated.

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl"
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/cyberscope/fdc
c/policy/?policy_id=17012&ips=10.10.24.10&organisation_name1=Name1&o
rganisation_name2=Name2&organisation_name3=Name3"
```

Sample 3. Asset Groups Filter

Use the **as_ids** parameter to include only hosts in the specified asset groups. Multiple asset group IDs are comma separated.

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl"
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/cyberscope/fdc
c/policy/?policy_id=17012&ag_ids=397405&ips=10.10.25.70&organisatio
n_name1=Name1&organisation_name2=Name2&organisation_name3=Name3"
```

SCAP Global Results API

The “SCAP Global Results” API v2 (the resource **/api/2.0/fo/asset/host/cyberscope/**) creates a Cyberscope report using the SCAP scan data saved for all the SCAP policies in the subscription and also the automatic VM scan data saved in the subscription. You must enter IPs/ranges and/or asset group IDs as input. These parameters allow users to customize the required “OrganisationName” elements in the XML report: organisation_name1, organisation_name2, and organisation_name3.

The service uses SCAP scan data for all the SCAP policies in the subscription and reports this in the datapoint <sr:DataPoint id:"configuration_management_agency_deviations">. This datapoint will include multiple Benchmark Data sections, one for each policy. Also the service uses the automatic VM data for applicable IPs (IPs in SCAP policies) and reports this in the datapoint <sr:DataPoint id:"vulnerability_management_product_vulnerabilities">.

Permissions: Users have permission to run this API function when the SCAP module is enabled for the user's subscription. Sub-accounts (Unit Managers, Scanners, and Readers) will view only data for IP addresses that their accounts have access to.

Sample 1. Select Hosts by IP

Use the **ips** parameter to select hosts by IP/range. You can enter a single IP, multiple IPs and/or IP ranges using the **ips** parameter. Multiple entries are comma separated. (This parameter and/or **ag_ids** is required.)

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl"
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/cyberscope/?ip
s=10.10.24.52&organisation_name1=Name1&organisation_name2=Name2&organis
ation_name3=Name3"
```

Sample 2. Select Hosts by Asset Group

Use the **as_ids** parameter to select hosts by asset group ID. You can enter one or more asset group IDs. Multiple IDs are comma separated. (This parameter and/or **ips** is required.)

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl"
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/cyberscope/?ag
_ids=503424&organisation_name1=Name1&organisation_name2=Name2&organisati
on_name3=Name3"
```

It's possible to select hosts by entering a combination of IPs/ranges and asset group IDs.

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl"
"https://qualysapi.qualys.com/api/2.0/fo/asset/host/cyberscope/?ip
s=10.10.24.52,10.10.25.2-
10.10.25.255&ag_ids=503424,503430&organisation_name1=Name1&organisati
on_name2=Name2&organisation_name3=Name3"
```

SCAP ARF Report

The "SCAP ARF Report" API v2 (the resource **`/api/2.0/fo/compliance/scap/arf/`**) allows users to create a SCAP scan report in Asset Reporting Format (ARF), a requirement in the SCAP 1.2 Specifications from NIST.

Permissions: Users have permission to run this API function when the SCAP module is enabled for the user's subscription. Sub-accounts (Unit Managers, Scanners and Readers) must have the "Manage compliance" permission.

Input parameters:

Parameter	Description
scan_id={value}	(Required) The scan ID for a finished SCAP scan.
ips={value}	(Optional) Use this parameter if you want to include only certain IP addresses in the report. You can enter a single IP, multiple IPs and/or ranges. Multiple entries are comma separated.
ips_network_id={value}	(Optional and valid only when the Network Support feature is enabled and the policy has SCAP 1.2 content) Use this parameter to restrict the report's target to the IPs specified in the "ips" parameter ("ips_network_id" is valid only when "ips" is specified in the same request).

How do I find the scan ID? You'll see the scan ID in the Qualys user interface, when viewing SCAP scan results. In the scan results window's title bar you'll see the report URL with its ID number in the "id" parameter, like this:
https:///qualyguard.qualys.com/fo/report/fdcc/fdcc_scan_result.php?id=3362251

API Request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X POST -d
"scan_id=3362251&ips=10.10.10.1-10.10.10.10"
"https://qualysapi.qualys.com/api/2.0/fo/compliance/scap/arf/"
```

XML Output:

The XML output is compliant with the ARF 1.1 Schema. Show me this schema

SCAP Policy List

The "SCAP Policy List" API v2 (the resource **`/api/2.0/fo/compliance/fdcc_policy/`** with the parameter **`action=list`**) is used to view a list of SCAP policies visible to the user. Optional input parameters support filtering the policy list output.

Maximum Policies per API Request

A maximum of 1,000 SCAP policy records can be processed per request. If the requested list identifies more than 1,000 policies, then the XML output includes the <WARNING> element and instructions for making another request for the next batch of policy records.

User Permissions

User permissions are described below.

User Role	Permissions
Manager	View all SCAP policies in subscription. View asset group information for all asset groups assigned to policies.
Auditor	View all SCAP policies in subscription. View asset group information for all asset groups assigned to policies.
Unit Manager	View all SCAP policies in subscription, when the "Manage compliance" permission is turned on in the user account settings. View asset group information for asset groups assigned to SCAP policies, when the user has permission to view these asset groups.
Scanner	View all SCAP policies in subscription, when the "Manage compliance" permission is turned on in the user account settings.. View asset group information for asset groups assigned to SCAP policies, when the user has permission to view these asset groups.
Reader	View all SCAP policies in subscription, when the "Manage compliance" permission is turned on in the user account settings.. View asset group information for asset groups assigned to SCAP policies, when the user has permission to view these asset groups.

Request

URL. Use this URL to request an SCAP policy list, where <qualysapi.qualys.com> is the API server URL where your account is located. Replace the API server URL if your account is located on another platform.

```
https://qualysapi.qualys.com/api/2.0/fo/
compliance/fdcc_policy/
```

Method. The GET or POST access method may be used.

Authentication. Authentication is required for each API request. See Chapter 2, "Authentication Using the V2 APIs."

Parameters

The parameters used to make an SCAP policy list request are below.

Parameter	Description
action=list	(Required) The action type required to request a list of SCAP policies.
echo_request={**0** | 1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When unspecified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
details={**Basic** | All | None}	(Optional) Show the requested amount of host information for each host. A valid value is: Basic - (default) Includes all SCAP policy details except the asset group list and SCAP file list All - includes all SCAP policy details None - includes SCAP policy ID and title
ids={value}	(Optional) Show only certain SCAP policy IDs/ranges. One or more policy IDs/ranges may be specified. Valid host IDs are required. Multiple entries are comma separated. A policy ID range is specified with a hyphen (for example, 190-400).
id_min={value}	(Optional) Show only SCAP policies which have a minimum SCAP policy ID value. A valid SCAP policy ID is required.
id_max={value}	(Optional) Show only SCAP policies which have a maximum SCAP policy ID value. A valid SCAP policy ID is required.

XML Output

DTD for SCAP Policy List

An API request for an SCAP policy list returns XML output using the DTD which can be found at the following URL:

```
https://qualysapi.qualys.com/api/2.0/fo/
compliance/fdcc_policy/fdcc_policy_list_output.dtd
```

where <qualysapi.qualys.com> is the API server URL where your account is located.

The DTD for the SCAP policy list XML output is provided in Appendix C.

Sample SCAP Policy List

Sample SCAP policy list output (fragment) with **details=All** is below.

```
<!DOCTYPE POLICY_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/compliance/fdcc_policy/fd
cc_policy_list_output.dtd">

<FDCC_POLICY_LIST_OUTPUT>
   <RESPONSE>
    <DATETIME>2012-07-19T22:10:16Z</DATETIME>
    <FDCC_POLICY_LIST>
      <FDCC_POLICY>
        <ID>10235</ID>
        <TITLE><![CDATA[XP policy]]></TITLE>
        <DESCRIPTION><![CDATA[This benchmark has been created to
assist IT professionals, in particular Windows XP system
administrators and information security personnel, in effectively
securing Windows XP Professional SP2 systems.]]></DESCRIPTION>
        <BENCHMARK><![CDATA[FDCC-Windows-XP]]></BENCHMARK>
<BENCHMARK_PROFILE><![CDATA[federal_desktop_core_configuration_ver
sion_1.2.1.0]]></BENCHMARK_PROFILE>
        <BENCHMARK_STATUS_DATE>2009-04-
08T00:00:00Z</BENCHMARK_STATUS_DATE>
        <VERSION><![CDATA[v1.2.1.0]]></VERSION>
        <TECHNOLOGY><![CDATA[Windows XP Desktop]]></TECHNOLOGY>
        <NIST_PROVIDED><![CDATA[No]]></NIST_PROVIDED>
        <CREATED>
          <DATETIME>2012-07-18T23:03:35Z</DATETIME>
```

```
            <BY>USERNAME</BY>
          </CREATED>
          <LAST_MODIFIED>
            <DATETIME>2012-07-18T23:03:35Z</DATETIME>
            <BY>USERNAME</BY>
          </LAST_MODIFIED>
          <ASSET_GROUP_LIST>
            <ASSET_GROUP>
              <ID>414242</ID>
              <TITLE><![CDATA[10.10.10.40]]></TITLE>
            </ASSET_GROUP>
            <ASSET_GROUP>
              <ID>414942</ID>
              <TITLE><![CDATA[10 range]]></TITLE>
            </ASSET_GROUP>
            <ASSET_GROUP>
              <ID>419582</ID>
              <TITLE><![CDATA[10.10.10.29]]></TITLE>
            </ASSET_GROUP>
            <ASSET_GROUP>
              <ID>419702</ID>
              <TITLE><![CDATA[10.10.10.28-16-191]]></TITLE>
            </ASSET_GROUP>
          </ASSET_GROUP_LIST>
          <FDCC_FILE_LIST>
            <FDCC_FILE>
             <FILE_NAME><![CDATA[fdcc-winxp-xccdf.xml]]></FILE_NAME>

<FILE_HASH><![CDATA[0c1a49c4ca47187995b543cfdcf35783]]></FILE_HASH
>
            </FDCC_FILE>
            <FDCC_FILE>
              <FILE_NAME><![CDATA[fdcc-winxp-cpe-
oval.xml]]></FILE_NAME>

<FILE_HASH><![CDATA[f397b9068b3881ef2a35c948326e6e4e]]></FILE_HASH
>
            </FDCC_FILE>
            <FDCC_FILE>
              <FILE_NAME><![CDATA[fdcc-winxp-cpe-
dictionary.xml]]></FILE_NAME>
```

```
<FILE_HASH><![CDATA[333b9b03961c58e65263bc86b4e0cdef]]></FILE_HASH
>
          </FDCC_FILE>
          <FDCC_FILE>
            <FILE_NAME><![CDATA[fdcc-winxp-oval.xml]]></FILE_NAME>

<FILE_HASH><![CDATA[d1cf1f195bb58f295ca4b17dea2f99f0]]></FILE_HASH
>
          </FDCC_FILE>
          <FDCC_FILE>
            <FILE_NAME><![CDATA[fdcc-winxp-
patches.xml]]></FILE_NAME>

<FILE_HASH><![CDATA[4ae1b306344ef564c5da479a4a3d7f53]]></FILE_HASH
>
          </FDCC_FILE>
        </FDCC_FILE_LIST>
      </FDCC_POLICY>
      <FDCC_POLICY>
...
    <FDCC_POLICY_LIST>
...
<FDCC_POLICY_LIST_OUTPUT>
```

CHAPTER 8

Scan Authentication API

The Authentication API is used to manage authentication records for authenticated scanning (e.g. trusted scanning).

These topics are covered:

- About Authenticated Scanning
- User Permissions and Authentication Records
- Authentication Record List
- Authentication Record List by Type
- Unix / Cisco IOS Record
- Windows Record
- Oracle Record
- Oracle Listener Record
- SNMP Record
- IBM DB2 Record
- VMware Record
- Application Server Records
- HTTP Record
- MySQL Record
- MS SQL Record (compliance scans only)
- Authentication Vault

About Authenticated Scanning

Authenticated scanning (also referred to as trusted scanning) provides organizations with authenticated host-level configuration and security data, enabling users to manage vulnerabilities and policy compliance at a deeper level in their network. With user-supplied authentication credentials, the scanning engine is able to authenticate to target hosts and return important information, such as missing patches and current password settings, that can be used to ensure that each host on the network is in compliance with internal security policies.

Several authentication types are available. All authentication types may be configured at the subscription-level. When authentication is enabled, the service automatically attempts authentication to target hosts, based on user-supplied credentials, for all scans launched by Managers, Unit Managers and Scanners.

For vulnerability scans, trusted scanning is optional but recommended. For compliance scans, trusted scanning is required. To use this feature, you must first create authentication records containing authentication credentials, and then enable the authentication feature in an option profile.

Login Credentials for Authentication

Each authentication record requires login credentials for authenticating to target hosts. Several documents are available on the Resources section in the Qualys user interface to assist users with trusted scanning, including instructions for configuring user accounts on target hosts for authentication. When you are logged into your Qualys account, go to Help > Resources > Tips and Techniques and click the download link for the document that you're interested in.

Target Hosts for Authentication

For a create request, specify the **ips** parameter to add IPs to an authentication record. If you specify this parameter for an update request, the IPs you specify replace any existing IPs. For example, if the target record includes IP 10.10.10.1 and the update request includes the parameter **ips=10.10.10.20**, then IP 10.10.10.20 is saved in the record and IP 10.10.10.1 is removed.

Other parameters are available for an update request, allowing you to manage IP addresses on an ongoing basis. The **add_ips** parameter allows you to append IP addresses to an existing record, and the **remove_ips** parameter allows you to remove IP addresses from an existing record. (Note if both **add_ips** and **remove_ips** are included in the same request, the IPs in **add_ips** are added first before IPs in **remove_ips** are removed.)

Batch Update Requests

Multiple authentication records may be updated at the same time. The attributes specified in the update request are assigned to all specified records. Records are updated in order by ID number.

If the change being made can only be applied to a single record, then only the first record in the list is updated and subsequent records are not updated. For example, each authentication record must have a unique title. This means that you cannot assign the same title to multiple records. Also, each IP address may only be included in one record per authentication type. Thus you cannot add the same set of IPs to multiple records of the same type.

User Permissions and Authentication Records

The user permissions for managing authentication records are described below. See "Managing Authentication Records" in the online help for all the details.

View Authentication Record List

User Role	Permissions
Manager	View all authentication records in subscription.
Unit Manager	View authentication records which contain hosts in the user's business unit.
Scanner	View authentication records which contain hosts in the user's assigned asset groups.
Auditor Reader	No permissions.

Create Authentication Record

User Role	Permissions
Manager	Create authentication records for hosts in the subscription.
Unit Manager	Create authentication records for hosts in the user's business unit. The permission "create/edit authentication records" must be granted in the user's account.
Auditor Scanner Reader	No permissions.

Update/Delete Authentication Record

User Role	Permissions
Manager	Update and delete authentication records.
Unit Manager	Update and delete authentication records. The permission "create/edit authentication records/vaults" must be granted in the user's account. To edit a record, at least one host in the record must be in the user's business unit. To delete a record, all hosts in the record must also be in the user's business unit.
Auditor Scanner Reader	No permissions.

Authentication Record List

The **/api/2.0/fo/auth** resource with the parameter **action=list** is used to view a list of all authentication records visible to the user. The record ID for each authentication record is listed in the authentication record list output. The output shows the authentication record ID only. To see authentication record attributes for a particular record, use the type-specific record list function (see "Authentication Record List by Type"). Authentication is required for each API request. See Chapter 2, "Authentication Using the V2 APIs."

Maximum Authentication Records per API Request

A maximum of 1,000 authentication records can be processed per request. If the requested list identifies more than 1,000 authentication records, then the XML output includes the <WARNING> element and instructions for making another request for the next batch of records.

User Permissions

See "User Permissions and Authentication Records" for information.

Record List: Request

URL.Use the URL syntax below to request an authentication record list, where <qualysapi.qualys.com> is the API server URL where your account is located. Replace the API server URL if your account is located on another platform.

```
https://<qualysapi.qualys.com>/api/2.0/fo/auth
```

Method. The GET or POST access method may be used.

Authentication. Authentication is required for each API request. See Chapter 2, "Authentication Using the V2 APIs."

Record List: Parameters

The parameters below may be specified for an authentication record list request.

Parameter	Description
action=list	(Required) Specifies the list action type for the request.
echo_request={**0** | 1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.

Parameter	Description
title={value}	(Optional) Show only authentication records which have a certain string in the record title.
comments={value}	(Optional) Show only authentication records which have a certain string in the record comments.
ids={value}	(Optional) Show only authentication records with certain IDs and/or ID ranges. Multiple entries are comma separated. One or more IDs/ranges may be specified. An ID range entry is specified with a hyphen (for example, 3000-3250). Valid IDs are required.
id_min={value}	(Optional) Show only authentication records which have a minimum ID value. A valid ID is required.
id_max={value}	(Optional) Show only authentication records which have a maximum ID value. A valid ID is required.

Record List: XML Output

DTD for Authentication Record List Output: All Types

An API request for an authentication record list with all authentication types returns XML output using the DTD listed below. The value `<qualysapi.qualys.com>` is the API server URL where your account is located.

All authentication record types:

```
https://<qualysapi.qualys.com>/api/2.0/fo/auth/auth_records.dtd
```

Want to learn more? Please see Appendix D.

Sample Authentication Record List: All Types

Sample authentication record list output with all authentication types:

```
<AUTH_ LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2008-05-21T13:32:17Z</DATETIME>
    <AUTH_RECORDS>
      <AUTH_UNIX_RECORDS>
        <ID_SET>
          <ID_RANGE>17-41</ID_RANGE>
          <ID_RANGE>62-119</ID_RANGE>
        </ID_SET>
      </AUTH_UNIX_RECORDS>
      <AUTH_WINDOWS_RECORDS>
        <ID_SET>
          <ID_RANGE>1-6</ID_RANGE>
        </ID_SET>
      </AUTH_WINDOWS_RECORDS>
      <AUTH_ORACLE_RECORDS>
        <ID_SET>
          <ID>7</ID>
        </ID_SET>
      </AUTH_ORACLE_RECORDS>
      <AUTH_SNMP_RECORDS>
        <ID_SET>
          <ID>4114</ID>
          <ID_RANGE>4117-4121</ID_RANGE>
        </ID_SET>
      </AUTH_SNMP_RECORDS>
      <AUTH_IBM_DB2_RECORDS>
        <ID_SET>
          <ID>6</ID>
        </ID_SET>
      </AUTH_IBM_DB2_RECORDS>
    </AUTH_RECORDS>
  </RESPONSE>
</AUTH_LIST_OUTPUT>
```

Authentication Record List by Type

The **/api/2.0/fo/auth/<type>** resource with the parameter **action=list** is used to view a list of authentication records visible to the user for a specific authentication type (Unix, Windows etc). Authentication is required for each API request. See Chapter 2, "Authentication Using the V2 APIs."

Records in the XML output are sorted by record ID in ascending order. Optional input parameters allow you to set filters to restrict the record list output to records with certain record IDs. The user has the ability to select the amount of additional information to include for each authentication record in the output. By default, basic authentication record information is included, showing all record attributes except password.

Use the **details** input parameter to select another level of detail to be included in the record list output. Two parameter settings are available. Specify **details=All** to show basic authentication record information and a glossary with user information. Specify **details=None** to list the authentication record ID only.

Maximum Authentication Records per API Request

A maximum of 1,000 authentication records can be processed per request. If the requested list identifies more than 1,000 authentication records, then the XML output includes the <WARNING> element and instructions for making another request for the next batch of records.

User Permissions

See "User Permissions and Authentication Records" for information.

Record List by Type: Request

URL. Use the URL syntax below to request an authentication record list for a specific authentication record type:

```
https://<qualysapi.qualys.com>/api/2.0/fo/auth/<type>
```

where <qualysapi.qualys.com> is the API server URL where your account is located. Replace the API server URL if your account is located on another platform.

<type> is the authentication record type. You may specify one these types: unix (for Unix and Cisco IOS records), windows, oracle, oracle_listener, snmp, vmware, apache, ms_iis, ibm_websphere, http, mysql, ms_sql.

Method. The GET or POST access method may be used.

Authentication. Authentication is required for each API request. See Chapter 2, "Authentication Using the V2 APIs."

Record List by Type: Parameters

The parameters below may be specified for an authentication record list request.

Parameter	Description
action=list	(Required) Specifies the list action type for the request.
echo_request={**0** | 1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
title={value}	(Optional) Show only authentication records which have a certain string in the record title.
comments={value}	(Optional) Show only authentication records which have a certain string in the record comments.
details={**Basic** | All | None}	(Optional) Show the requested amount of information for each authentication record. A valid value is: None — show record ID only Basic (default) — show record ID and all authentication record attributes All — show record ID and all authentication record attributes and a glossary section with the user name and login for each record owner
ids={value}	(Optional) Show only authentication records with certain IDs and/or ID ranges. Multiple entries are comma separated. One or more IDs/ranges may be specified. An ID range entry is specified with a hyphen (for example, 3000-3250). Valid IDs are required.
id_min={value}	(Optional) Show only authentication records which have a minimum ID value. A valid ID is required.
id_max={value}	(Optional) Show only authentication records which have a maximum ID value. A valid ID is required.

Record List by Type: XML Output

DTD for Authentication Record List Output

An API request for an authentication record list returns XML output using the DTD listed below. The value <qualysapi.qualys.com> is the API server URL where your account is located.

Specific authentication record types:

```
https://<qualysapi.qualys.com>/api/2.0/fo/auth/<type>/
auth_<type>_list_output.dtd
```

where <type> is the authentication record type, such as unix, windows, oracle, etc.

Want to learn more? Please see Appendix D.

Sample Authentication Record List: Specific Type

Sample authentication record list output listing Unix and Cisco IOS records:

```
<AUTH_UNIX_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2008-05-21T13:32:17Z</DATETIME>
    <AUTH_UNIX_LIST>
      <AUTH_UNIX>
        <ID>678</ID>
        <TITLE><![CDATA[My Ubuntu credentials]]></TITLE>
        <USERNAME><![CDATA[bumbler]]></USERNAME>
        <ROOT_TOOL>Sudo</ROOT_TOOL>
        <CLEARTEXT_PASSWORD>0</CLEARTEXT_PASSWORD>
        <IP_SET>
          <IP_RANGE>10.10.10.168-10.10.10.195</IP_RANGE>
        </IP_SET>
        <CREATED>
            <DATETIME>2008-04-20T01:01:01</DATETIME>
            <BY>quays_es11</BY>
        </CREATED>
        <LAST_MODIFIED>
            <DATETIME>2008-04-20T01:01:01</DATETIME>
            <BY>quays_es11</BY>
        </LAST_MODIFIED>
        </COMMENTS><![CDATA[Development lab]]></COMMENTS>
      </AUTH_UNIX>
      ...
    </AUTH_UNIX_LIST>
    <WARNING_LIST>
        <WARNING>
            <CODE>1980</CODE>
            <TEXT>1000 record limit exceeded. Use URL to get next
batch of records.</TEXT>

<URL>https://qualysapi.qualys.com/api/2.0/fo/auth/?action=list&id_
min=3457</URL>
        </WARNING>
    </WARNING_LIST>
```

```
    <GLOSSARY>
        <USER_LIST>
            <USER>
                <USER_LOGIN>quays_es11</USER_LOGIN>
                <FIRST_NAME>Ernie</FIRST_NAME>
                <LAST_NAME>Smith</LAST_NAME>
            </USER>
        </USER_LIST>
    </GLOSSARY>
  </RESPONSE>
</AUTH_UNIX_LIST_OUTPUT>
```

Unix / Cisco IOS Record

The **/api/2.0/fo/auth/unix** resource allows you to manage Unix and Cisco IOS authentication records for authenticated scanning. You can submit API requests to view Unix and Cisco IOS records, add new records, update records and delete records. Authentication is required for each API request. See Chapter 2, "Authentication Using the V2 APIs."

Maximum Authentication Records per API Request

A maximum of 1,000 authentication records can be processed per request, when requesting records to view, update or delete in batch mode. If the requested list identifies more than 1,000 authentication records, then the XML output includes the <WARNING> element and instructions for making another request for the next batch of records.

User Permissions

See "User Permissions and Authentication Records" for information.

Unix/Cisco IOS Record: Request

URL. Use this URL to make a Unix/Cisco IOS record request (create, update, or delete), where <qualysapi.qualys.com> is the API server URL where your account is located. Replace the API server URL if your account is located on another platform.

```
https://<qualysapi.qualys.com>/api/2.0/fo/auth/unix
```

Method. For a list request, the GET or POST method may be used. For another request (create, update, or delete), the POST access method must be used.

Authentication. Authentication is required for each API request. See Chapter 2, "Authentication Using the V2 APIs."

Unix/Cisco IOS Record: List

Specify a request that returns a list of Unix and Cisco IOS authentication records. See "Authentication Record List by Type" for details.

Unix/Cisco Record: Create

The parameters described below are used to specify a request that creates a new Unix or Cisco IOS authentication record.

Parameter	Description
action=create	(Required) The action type for the API request: create. To create a Unix or Cisco IOS record, you must include **action=create** with the API request.
sub_type=cisco	(Required for Cisco IOS record; Invalid for Unix record) To create a Cisco IOS record, you must include **sub_type=cisco** with the API request.
echo_request={**0**|1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to include parameters in the XML output.
title={value}	(Required) Specifies a title for the authentication record. The title must be unique and may include a maximum of 255 characters (ascii).
comments={value}	(Optional) Specifies user defined notes about the authentication record. The comments may include a maximum of 1999 characters (ascii); if comments have 2000 or more characters an error is returned and comments are not saved. Tags (such as <script>) cannot be included; if tags are included an error is returned and the request fails.
{login credentials}	Define login credentials for authentication to target hosts. See "Unix/Cisco IOS Record Properties: Login Credentials."
enable_password={value}	(Optional for Cisco IOS record; Invalid for Unix record) The password required for executing the "enable" command on the target hosts. The password may include 1-31 characters (ascii). Note: The pooled credentials feature is not supported if the "enable" command requires a password and it is specified using the **enable_password** parameter.
port={value}	(Optional for compliance scanning: Invalid for vulnerability scanning) Specifies custom ports to be used to perform authentication and compliance assessment (control testing). See "Unix/Cisco IOS Record Properties: Custom Ports."
{target hosts}	Define the target hosts for authentication. See "Unix/Cisco IOS Record Properties: Target Hosts."
use_agentless_tracking=[**0**|1}	Specify "1" to enable Agentless Tracking.

Parameter	Description
agentless_tracking_path={value}	(Valid for a Unix record only). The pathname where you would like the service to store the host ID file on each host. This is required to enable Agentless Tracking for a Unix record; this is not valid for a Windows record.
network_id={value}	(Optional and valid when the networks feature is enabled) The network ID for the record.

Unix/Cisco IOS Record: Update

The parameters described below are used to specify a request to update one or more existing Unix and/or Cisco IOS authentication records. Parameters specified in an update request will overwrite any existing parameters (previously defined).

Parameter	Description
action=update	(Required) The action type for the API request: update.
echo_request={**0** | 1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
ids={value}	(Required) Update only authentication records with certain IDs and/or ID ranges. Multiple entries are comma separated. One or more IDs/ranges may be specified. An ID range entry is specified with a hyphen (for example, 1359-1407). Valid IDs are required.
title={value}	(Optional) Overwrites the existing record title with a new title. The title must be unique and may include a maximum of 255 characters (ascii). When multiple IDs are specified for a batch update, the title is only replaced in the first record ID in the action since the title must be unique across records. The update will fail for the remaining IDs included in the action.
comments={value}	(Optional) Specifies user defined notes about the authentication record. The comments may include a maximum of 1999 characters (ascii); if comments have 2000 or more characters an error is returned and comments are not saved. Tags (such as <script>) cannot be included; if tags are included an error is returned and the request fails.
{login credentials}	Define login credentials for authentication to target hosts. See "Unix/Cisco IOS Record Properties: Login Credentials."

Parameter	Description
enable_password={value}	(Optional for Cisco IOS record; Invalid for Unix record) The password required for executing the "enable" command on the target hosts. The password may include 1-31 characters (ascii). Note: The pooled credentials feature is not supported if the "enable" command requires a password and it is specified using the **enable_password** parameter.
port={value}	(Optional for compliance scanning: Invalid for vulnerability scanning) Specifies custom ports to be used to perform authentication and compliance assessment (control testing). See "Unix/Cisco IOS Record Properties: Custom Ports."
{target hosts}	Define the target hosts for authentication. See "Unix/Cisco IOS Record Properties: Target Hosts."
use_agentless_tracking=[**0**|1}	Specify "1" to enable Agentless Tracking.
agentless_tracking_path={value}	(Valid for a Unix record only). The pathname where you would like the service to store the host ID file on each host. This is required to enable Agentless Tracking for a Unix record; this is not valid for a Windows record.
network_id={value}	(Optional and valid when the networks feature is enabled) The network ID for the record.

Unix/Cisco IOS Record: Delete

The parameters described below are used to specify a request that deletes one or more Unix and/or Cisco IOS authentication records.

Parameter	Description
action=delete	(Required) The action type for the API request: delete.
echo_request={**0**|1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
ids={value}	(Required) Delete only authentication records with certain IDs and/or ID ranges. Multiple entries are comma separated. One or more IDs/ranges may be specified. An ID range entry is specified with a hyphen (for example, 3000-3250). Valid IDs are required.

Unix/Cisco IOS Record Properties: Login Credentials

The following parameters are used to define the login credentials for the account to be used for authentication. Parameters specified in an update request will overwrite any existing parameters (previously defined).

For user account requirements, see "Unix Authentication Setup" and "Cisco IOS Authentication Setup" in the Qualys online help.

For Unix authentication, the service attempts authentication to hosts that support SSH2, in the following order: 1) RSA key, 2) DSA key and 3) password. The service attempts authentication to hosts that only support SSH1 using the supplied user name and password.

Parameter	Description
username={value}	(Required for create request; Optional for update request) Specifies the user account to be used for authentication on target hosts. The username may include 1-31 characters (ascii).
root_tool= {**None** | Sudo | PowerBroker}	(Optional for Unix record; Invalid for Cisco IOS record) Specifies the root delegation tool to be used. When a root delegation tool (Sudo or PowerBroker) is specified, the service authenticates to target hosts using a lower privileged account and performs scan tests using the elevated privileges of the superuser (root). When unspecified, a root delegation tool will not be used. BeyondTrust PowerBroker version 6 is supported. See "Enable Root Delegation" and "PowerBroker Integration" in the online help for further information and requirements.
password={value}	(Required for create request only if the **cleartext_password** parameter is also specified; Optional for update request) Specifies the password corresponding to the user account defined in the record for authentication. The password may include 1-31 characters (ascii). For a Unix record: If you plan to supply an RSA and/or DSA private key for authentication then a password is not required. However, supplying a password is recommended in case key authentication fails.

Parameter	Description
cleartext_password={**0** | 1}	(Optional) When not specified, the scanning engine only uses strong password encryption for remote login. Specify 1 to allow your password to be transmitted in clear text when connecting to services which do not support strong password encryption. For more information, see "Clear Text Password" in the online help or refer to the *Unix Trusted Scanning* document (log into your account and go to Help > Resources > Tips and Techniques).
	For a create request, if the **cleartext_password** parameter is specified, then the **password** parameter must also be specified. For an update request, if the **cleartext_password** parameter is specified and the record does not have a password set, then this parameter is *silently ignored*.
rsa_private_key={value}	(Optional for Unix record; Invalid for Cisco IOS record) Specifies an RSA private key for authentication. Copy the contents of an RSA private key file (id_rsa) and be sure to include the first and last line (-----BEGIN RSA PRIVATE KEY----- and -----END RSA PRIVATE KEY-----). You may include an X.509 certificate with your private key. Enter the certificate block after the key block and be sure to include the first and last line (-----BEGIN CERTIFICATE----- and -----END CERTIFICATE-----). Enter a maximum of 128 characters (ascii).
dsa_private_key={value}	(Optional for Unix record; Invalid for Cisco IOS record) Specifies a DSA private key for authentication. Copy the contents of a DSA private key file (id_dsa) and be sure to include the first and last line (-----BEGIN DSA PRIVATE KEY----- and -----END DSA PRIVATE KEY-----). You may include an X.509 certificate with your private key. Enter the certificate block after the key block and be sure to include the first and last line (-----BEGIN CERTIFICATE----- and -----END CERTIFICATE-----). Enter a maximum of 128 characters (ascii).

Unix/Cisco IOS Record Properties: Custom Ports

For Unix and Cisco IOS compliance scanning Qualys users have the option to select custom ports to be used to authenticate to target hosts and perform compliance assessment (control testing). The custom ports option allows users to scan hosts for compliance when services (SSH, telnet, rlogin) are not running on well known ports (22, 23, 513 respectively). A "port" input parameter is used to define a custom ports list.

The service automatically performs compliance scanning using these well known ports: 22 (SSH), 23 (telnet) and 513 (rlogin). Any one of these services (SSH, telnet, rlogin) is sufficient for compliance scanning. Depending on your network you may need to define custom ports for Unix and Cisco IOS compliance scanning to be successful.

Ports Used For Unix/Cisco IOS Compliance Scanning

The actual ports used for Unix and Cisco IOS compliance scanning depends on these scan settings: 1) compliance option profile, and 2) Unix or Cisco IOS authentication record.

Compliance Profile (UI only)	Authentication Record	Ports Scanned
Standard Scan	UI: Well Known Ports API: no "port" parameter	~ 1900 Ports (includes Ports 22, 23, 513)
Standard Scan	UI: Custom Ports API: "port" parameter	~ 1900 Ports + Custom Ports in record
Targeted Scan	UI: Well Known Ports API: no "port" parameter	Ports 22, 23 and 513 only
Targeted Scan	UI: Custom Ports API: "port" parameter	Custom Ports in record

Ports Setting in Compliance Profile

A Ports setting is available in the compliance option profile for compliance scanning. The compliance profile may be created/edited using the Qualys user interface only.

Note: The Ports setting in the compliance profile applies to Unix, Cisco IOS and Windows scans only. For other scan types, including Oracle, MS SQL and SNMP, the Ports setting does not apply. For Oracle and MS SQL scan types, the service always scans the ports defined in the corresponding authentication records.

These Ports settings are available:

Standard Scan. When selected, the standard ports (about 1900 ports) are used for compliance scanning, including the well known ports: 22 (SSH), 23 (telnet) and 513 (rlogin).

Targeted Scan. When selected, the service targets the scan to a smaller set of ports than the standard ports (about 1900 ports). This is the recommended setting, and it is the initial setting for a new compliance profile. For Unix and Cisco IOS hosts, these well known ports are scanned: 22 (SSH), 23 (telnet) and 513 (rlogin). Any one of these services is sufficient for authentication. If services (SSH, telnet, rlogin) are not running on these well known ports for the hosts you will be scanning, select this option and define a

custom ports list in the authentication record. Note: The actual ports scanned also depends on the Ports setting in the authentication record. See "Ports Used For Unix/Cisco IOS Compliance Scanning."

Ports Setting in Unix/Cisco IOS Record

A Ports setting is available in the Unix and Cisco IOS record allowing users to select custom ports for compliance scanning (authentication and compliance assessment).

These Ports settings are available:

Well Known Ports. The service scans services (SSH, telnet, rlogin) on well known ports (22, 23, 513 respectively). Using the authentication record API, this is the default behavior.

Custom Ports. When the "port" input parameter is specified, the service scans a custom list of ports. Select this option if services (SSH, telnet, rlogin) are not running on well known ports (22, 23, 513 respectively). Note: The actual ports scanned also depends on the Ports setting in the compliance option profile. See "Ports Used For Unix/Cisco IOS Compliance Scanning."

"port" Input Parameter

(Optional when the Policy Compliance module is enabled in the user's account)

Using the "port" parameter you can add a custom ports list to a new or existing Unix or Cisco IOS authentication record. The "port" parameter is required if services (SSH, telnet, rlogin) are not running on well known ports (22, 23, 513 respectively). The service requires one of these services to scan a host for compliance.

Parameter	Description
port={value}	(Optional for compliance scans; Invalid for vulnerability scans) Specifies custom ports to be used for compliance scanning (authentication and compliance assessment). One or more ports may be specified. Multiple ports are comma separated. A maximum of 128 characters may be specified. Note: When custom ports are specified, the actual ports scanned also depends on the Ports setting in the compliance profile for any given compliance scan. See "Ports Used For Unix/Cisco IOS Compliance Scanning."

Using the Qualys user interface, users have the option to define custom ports in a Unix or Cisco IOS record in the "Custom Ports" field.

Unix/Cisco IOS Record Properties: Target Hosts

The following parameters are used to define the target hosts the scanning engine should log into with the specified credentials.

Important Note About IPs: Each IP address in your account may be included in one Unix or one Cisco IOS record within one Qualys user account. For example IP address 10.10.10.100 cannot be added to both a Unix record and a Cisco IOS record within your Qualys account.

Parameter	Description
ips={value}	(Required for create request) Add IP address(es) to the IP list for a new authentication record. You may enter a combination of IPs and IP ranges. Multiple entries are comma separated. See "Important Note About IPs" above. (Optional for update request) Overwrites (replaces) the IP address(es) in the IP list for an existing authentication record. The IPs you specify are added, and any existing IPs are removed. You may enter a combination of IPs and IP ranges. Multiple entries are comma separated.
	This parameter cannot be specified in the same request with the **`add_ips`** parameter or the **`remove_ips`** parameter.
add_ips={value}	(Optional for update request: Invalid for create request) Add IP address(es) to the IP list for ran existing authentication record. You may enter a combination of IPs and IP ranges. Multiple entries are comma separated. See "Important Note About IPs" above.
	This parameter and the **`ips`** parameter cannot be specified in the same request.
remove_ips={value}	(Optional for update request, Invalid for create request) Remove IP address(es) from the IP list for an existing authentication record. You may enter a combination of IPs and IP ranges. Multiple entries are comma separated. See "Important Note About IPs" above.
	This parameter and the **`ips`** parameter cannot be specified in the same request.

Unix/Cisco IOS Record: Sample API Requests

These sample requests work on Qualys US Platform 1 where the FQDN in the API server URL is qualysapi.qualys.com. Please be sure to replace the FQDN with the proper API server URL for your platform. For the EU platform, use qualysapi.qualys.eu. For a partner platform, use the URL for your @customer platform API server.

The header parameter "X-Requested-With" is also provided as an example. Please change this parameter according to your need. This parameter is exposed in the activity log (only displayed when used with session cookie authentication - does not apply for basic authentication as used in the following examples) and it is useful for API monitoring API activity.

Sample 1. Use the URL shown below to create a new Cisco IOS record:

```
curl -k -H 'X-Requested-With: Curl' -u 'username:password' -X "POST"
"https://qualysapi.qualys.com/api/2.0/fo/auth/unix/?action=create&sub
_type=cisco&title=My+Cisco+IOS+Record&username=John&password=abc12345
&cleartext_password=1&enable_password=chocolate&ips=10.10.10.10-
10.10.10.12"
```

Sample 2. Use the URL shown below to update the record ID 333 (Unix or Cisco IOS record) by adding new IP addresses:

```
curl -k -H "X-Requested-With: Curl" -u "username:password" -X "POST"
"https://qualysapi.qualys.com/api/2.0/fo/auth/unix/?action=update&ids
=333&add_ips=10.10.10.200-10.10.10.240"
```

Sample 3. Use the URL shown below to delete record ID 55 (Unix or Cisco IOS record):

```
curl -k -H "X-Requested-With: Curl" -u "username:password" -X "POST"
"https://qualysapi.qualys.com/api/2.0/fo/auth/unix/?action=delete&ids
=55"
```

Sample 4. Use the URL shown below to create a Unix record with Agentless Host Tracking:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d
"action=create&echo_request=1&title=UnixAuthRecord&comments=API+Testi
ng&username=root&password=abc123&ips=10.10.26.2&use_agentless_trackin
g=1&agentless_tracking_path=/etc/tmp/"
"https://qualysapi.qualys.com/api/2.0/fo/auth/unix/" >
CreateUnixRecordWithAgentless.xml
```

Unix/Cisco IOS Record: XML Output

The XML output returned by a Unix/Cisco IOS record API request (create, edit or delete) uses the DTD at the following URL (where qualysapi.qualys.com is the API server URL where your account is located):

```
https://qualysapi.qualys.com/api/2.0/batch_return.dtd
```

The batch_return.dtd for the XML output is provided in Appendix D.

Windows Record

The **/api/2.0/fo/auth/windows** resource allows you to manage Windows authentication records for authenticated scanning. You can submit API requests to view Windows authentication records, add new records, update records and delete records. Authentication is required for each API request. See Chapter 2, "Authentication Using the V2 APIs."

Maximum Authentication Records per API Request

A maximum of 1,000 authentication records can be processed per request, when requesting records to view, update or delete in batch mode. If the requested list identifies more than 1,000 authentication records, then the XML output includes the <WARNING> element and instructions for making another request for the next batch of records.

User Permissions

See "User Permissions and Authentication Records" for information.

Before You Begin

Before you create authentication records, you must configure authentication credentials on target hosts. Please refer to the *Windows Trusted Scanning* document for information, including account requirements and setup instructions. This document may be downloaded (in PDF) when you are logged into your Qualys account. Go to Help > Resources > Tips and Techniques and click the download link provided.

Windows Record: Request

URL. Use this URL to make a Windows record request (list, create, update, or delete), where <qualysapi.qualys.com> is the API server URL where your account is located. Replace the API server URL if your account is located on another platform.

```
https://<qualysapi.qualys.com>/api/2.0/fo/auth/windows
```

Method. For a list request, the GET or POST method may be used. For another request (create, update, or delete), the POST access method must be used.

Authentication. Authentication is required for each API request. See Chapter 2, "Authentication Using the V2 APIs."

Windows Record: List

Specify a request that returns a list of Windows authentication records. See "Authentication Record List by Type" for details.

Windows Record: Create

The parameters described below are used to specify a request that creates a new Windows authentication record.

Parameter	Description
action=create	(Required) Specifies the create action type for the request.
echo_request={**0**|1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
title={value}	(Required) Specifies a title for the authentication record. The title must be unique and may include a maximum of 255 characters (ascii).
comments={value}	(Optional) Specifies user defined notes about the authentication record. The comments may include a maximum of 1999 characters (ascii); if comments have 2000 or more characters an error is returned and comments are not saved. Tags (such as <script>) cannot be included; if tags are included an error is returned and the request fails.
{login credentials}	Define login credentials for authentication to target hosts. See "Windows Record Properties: Login Credentials."
{target hosts}	Define the target hosts for authentication. See "Windows Record Properties: Target Hosts."
use_agentless_tracking=[**0**|1}	Specify "1" to enable Agentless Tracking.
network_id={value}	(Optional and valid when the networks feature is enabled) The network ID for the record.

Windows Record: Update

The parameters described below are used to specify a request to update one or more existing Windows authentication records. Parameters specified in an update request will overwrite any existing parameters (previously defined).

Parameter	Description
action=update	(Required) Specifies the update action type for the request.
echo_request={**0**|1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.

Parameter	Description
ids={value}	(Required) Update only authentication records with certain IDs and/or ID ranges. Multiple entries are comma separated. One or more IDs/ranges may be specified. An ID range entry is specified with a hyphen (for example, 1359-1407). Valid IDs are required.
title={value}	(Optional) Overwrites the existing record title with a new title. The title must be unique and may include a maximum of 255 characters (ascii). When multiple IDs are specified for a batch update, the title is only replaced in the first record ID in the action since the title must be unique across records. The update will fail for the remaining IDs included in the action.
comments={value}	(Optional) Specifies user defined notes about the authentication record. The comments may include a maximum of 1999 characters (ascii); if comments have 2000 or more characters an error is returned and comments are not saved. Tags (such as <script>) cannot be included; if tags are included an error is returned and the request fails.
{login credentials}	Define login credentials for authentication to target hosts. See "Windows Record Properties: Login Credentials."
{target hosts}	Define the target hosts for authentication. See "Windows Record Properties: Target Hosts."
use_agentless_tracking=[**0**|1}	Specify "1" to enable Agentless Tracking.
network_id={value}	(Optional and valid when the networks feature is enabled) The network ID for the record.

Windows Record: Delete

The parameters described below are used to specify a request that deletes one or more Unix authentication records.

Parameter	Description
action=delete	(Required) Specifies the delete action type for the request.
echo_request={**0**|1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
ids={value}	(Required) Delete only authentication records with certain IDs and/or ID ranges. Multiple entries are comma separated. One or more IDs/ranges may be specified. An ID range entry is specified with a hyphen (for example, 3000-3250). Valid IDs are required.

Windows Record Properties: Login Credentials

When defining login credentials for Windows authentication, you must supply a username and password for a Windows account.

Interested in domain level authentication? These domain types are supported: Active Directory, NetBIOS User-Selected IPs, NetBIOS Service-Selected IPs. Use the parameter "windows_ad_domain" for Active Directory and "windows_domain" for NetBIOS.

A few notes:

- Once a record is saved, you cannot change the domain type from Active Directory to NetBIOS or from NetBIOS to Active Directory.
- Authentication is performed at the local host level when neither of these parameters are specified: **windows_ad_domain** and **windows_domain**

The following parameters are used to define the login credentials. Parameters specified in an update request will overwrite any existing parameters (previously defined).

Parameter	Description
username={value}	(Required for create request; Optional for update request) Specifies the user account to be used for authentication on target hosts. The username may include 1-31 characters (ascii).
password={value}	(Required for create request; Optional for update request) Specifies the password corresponding to the user account defined in the record for authentication. The password may include 1-31 characters (ascii).
windows_ad_domain={value}	(Optional) Specifies a Windows Active Directory domain name for domain level authentication. When specified, we'll use an Active Directory forest to authenticate to hosts in a certain domain within the framework. You'll need to enter a Fully Qualified Domain Name (FQDN).
	This parameter and the **windows_domain** parameter cannot be specified in the same request.
	This parameter and the **ips** parameter cannot be specified in the same request.

Parameter	Description
windows_domain={value}	(Optional) Specifies a Windows NetBIOS domain name for domain level authentication. This parameter and the **windows_ad_domain** parameter cannot be specified in the same request. When the **ips** parameter is also specified, the domain type is NetBIOS, User-Selected IPs. We'll use NetBIOS to authenticate to the IPs in the domain configuration. When the **ips** parameter is not specified, the domain type is NetBIOS, Service-Selected IPs. We'll use NetBIOS to authenticate to hosts in the domain using credentials stored on the domain.
ntlm={0|1}	(Optional) When not specified, NTLM authentication is enabled allowing the scanning engine to try the NTLM authentication protocol when negotiating authentication to target hosts. Specify **ntlm=0** if you do not want the NTLM authentication protocol attempted for the hosts defined in the Windows record. This may be the case if the target hosts are running a version of Windows that supports a more secure authentication protocol like Kerberos. When NTLM authentication is disabled, it will not be attempted even if other methods like NTLMSSP and Kerberos fail.

Windows Record Properties: Target Hosts

Each IP address in your account may be included in one Windows record.

The following parameters are used to define the target hosts that the scanning engine should log into with the specified credentials.

Parameter	Description
ips={value}	(Optional for create request) When specified, the **ips** parameter defines an IP list for the authentication record. These are the hosts to be authenticated to with the login credentials. When not specified, the service uses the authentication credentials stored on the domain to authenticate to hosts that are members of the domain. (Optional for update request) Overwrites (replaces) the IP list for the authentication record. The IPs you specify are added and any existing IPs are removed. You may enter a combination of IPs and IP ranges. Multiple entries are comma separated.
	This parameter and the **add_ips** parameter or the **remove_ips** parameter cannot be specified in the same request.
add_ips={value}	(Optional for update request only) This parameter is used to update an existing IP list in an existing authentication record. Specifies one or more IP addresses to add to the IP list for the authentication record. You may enter a combination of IPs and IP ranges. Multiple entries are comma separated.
	This parameter and the **ips** parameter cannot be specified in the same request.
remove_ips={value}	(Optional for update request only) This parameter is used to update an existing IP list in an existing authentication record. Specifies one or more IP addresses to remove from the IP list for the authentication record. You may enter a combination of IPs and IP ranges. Multiple entries are comma separated.
	This parameter and the **ips** parameter cannot be specified in the same request.

Windows Record: XML Output

The XML output returned by a Windows record API request (create, edit or delete) uses the DTD at the following URL (where qualysapi.qualys.com is the API server URL where your account is located):

```
https://qualysapi.qualys.com/api/2.0/batch_return.dtd
```

Want to learn more? Please see Appendix D.

Oracle Record

The **/api/2.0/fo/auth/oracle** resource allows you to manage Oracle authentication records for authenticated scanning. You can submit API requests to view Oracle authentication records, add new records, update records and delete records. Authentication is required for each API request. See Chapter 2, "Authentication Using the V2 APIs."

During scanning the service will authenticate to one or more Oracle instances on a single host using all the Oracle records in your account. For compliance scans, the user can allow the service to authenticate to multiple Oracle instances on a single host and port combination.

See the topic "Oracle Use Cases" in the Qualys online help to learn more about how to define Oracle authentication records for various network configurations.

Maximum Authentication Records per API Request

A maximum of 1,000 authentication records can be processed per request, when requesting records to view, update or delete in batch mode. If the requested list identifies more than 1,000 authentication records, then the XML output includes the <WARNING> element and instructions for making another request for the next batch of records.

User Permissions

See "User Permissions and Authentication Records" for information.

Before You Begin

The service provides a collection of scripts for successfully setting up Oracle trusted scanning for vulnerability scans and compliance scans. The scripts guide you through creating a user account with required privileges for authenticated scanning. Scripts and step-by-step instructions are described in the following documents:

Oracle Trusted Scanning: Setup for Vulnerability Scans

Oracle Trusted Scanning: Setup for Compliance Scans

Both documents may be downloaded (in PDF) when you are logged into your Qualys account. Go to Help > Resources > Tips and Techniques and click the download link provided.

Oracle Record: Request

URL. Use this URL to make an Oracle authentication record request (create, update, or delete), where `<qualysapi.qualys.com>` is the API server URL where your account is located. Replace the API server URL if your account is located on another platform.

```
https://<qualysapi.qualys.com>/api/2.0/fo/auth/oracle
```

Method. For a list request, the GET or POST method may be used. For another request (create, update, or delete), the POST access method must be used.

Authentication. Authentication is required for each API request. See Chapter 2, "Authentication Using the V2 APIs."

Oracle Record: List

Specify a request that returns a list of Oracle authentication records. See "Authentication Record List by Type" for details.

Oracle Record: Create

The parameters below are used for a request to create a new Oracle authentication record.

Parameter	Description
action=create	(Required) Specifies the create action type for the request.
echo_request={**0**|1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
title={value}	(Required) A title for the authentication record. The title must be unique and may include a maximum of 255 characters (ascii).
comments={value}	(Optional) User defined notes about the authentication record. A maximum of 1999 characters (ascii) may be specified; if comments have 2000 or more characters an error is returned and comments are not saved. Tags (such as <script>) cannot be included; if tags are included an error is returned and the request fails.
{login credentials}	Define login credentials for authentication to target hosts. See "Oracle Record Properties: Login Credentials."
{target hosts}	Define the target hosts for authentication. See "Oracle Record Properties: Target Hosts."

Parameter	Description
{OS properties}	Define operating system properties for compliance scans. See "Oracle Record Properties: OS Parameters."
network_id={value}	(Optional and valid when the networks feature is enabled) The network ID for the record.

Oracle Record: Update

The parameters described below are used to specify a request to update one or more existing Oracle authentication records. Parameters specified in an update request will overwrite any existing parameters (previously defined).

Parameter	Description
action=update	(Required) Specifies the update action type for the request.
echo_request={**0** | 1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
ids={value}	(Required) Update only authentication records with certain IDs and/or ID ranges. Multiple entries are comma separated. One or more IDs/ranges may be specified. An ID range entry is specified with a hyphen (for example, 1359-1407). Valid IDs are required.
title={value}	(Optional) Overwrites the existing record title with a new title. The title must be unique and may include a maximum of 255 characters (ascii). When multiple IDs are specified for a batch update, the title is only replaced in the first record ID in the action since the title must be unique across records. The update will fail for the remaining IDs included in the action.
comments={value}	(Optional) User defined notes about the authentication record. The comments may include a maximum of 1999 characters (ascii); if comments have 2000 or more characters an error is returned and comments are not saved. Tags (such as <script>) cannot be included; if tags are included an error is returned and the request fails.
{login credentials}	Define login credentials for authentication to target hosts. See "Oracle Record Properties: Login Credentials."
{target hosts}	Define the target hosts for authentication. See "Oracle Record Properties: Target Hosts."

Parameter	Description
{OS properties}	Define operating system properties for compliance scans. See "Oracle Record Properties: OS Parameters."
network_id={value}	(Optional and valid when the networks feature is enabled) The network ID for the record.

Oracle Record: Delete

The parameters described below are used to specify a request that deletes one or more Oracle authentication records.

Parameter	Description
action=delete	(Required) Specifies the delete action type for the request.
echo_request={**0**|1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
ids={value}	(Required) Delete only authentication records with certain IDs and/or ID ranges. Multiple entries are comma separated. One or more IDs/ranges may be specified. An ID range entry is specified with a hyphen (for example, 3000-3250). Valid IDs are required.

Oracle Record Properties: Login Credentials

The following parameters are used to define the login credentials. Parameters specified in an update request are optional and will overwrite any existing parameters (previously defined).

Parameter	Description
username={value}	(Required to create a record) The user account to be used for authentication to the Oracle database. The username may include 1-31 characters (ascii).
password={value}	(Required to create a record) The password corresponding to the user account defined in the record for authentication. The password may include 1-31 characters (ascii).
sid={value}	(To create a new record **sid** or **servicename** is required) The Oracle System ID (SID) that identifies the database instance to be authenticated to. The parameters **sid** and **servicename** cannot be specified in the same request.

Parameter	Description
servicename={value}	(To create a new record **sid** or **servicename** is required) The Oracle service name that identifies the database instance to be authenticated to. A maximum of 30 characters may be specified.
	The parameters **sid** and **servicename** cannot be specified in the same request.
port={value}	(Optional to create a record) The port number that the Oracle database instance is running on. When not specified, the "All Ports" option is used and the scanning engine will authenticate to the database instance on each port that the Oracle service is detected on. See "All Ports" below for information.
pc_only={**0** | 1)	(Optional to create a record, valid when the compliance module is enabled) Specify 1 to perform compliance scans on multiple instances running on host and port combinations in this record. This parameter must be specified if this Oracle record has some host and port combination, which is already defined in another record. Note, however, when **pc_only=1** is specified, the record will be used for compliance scans only. When not specified, the record will be used for vulnerability scans and compliance scans.

All Ports

The "All Ports" option is used when the **port** parameter is not specified (the default). When specified, the scanning engine uses the credentials in the record to attempt authentication to the database instance (SID or service name) when a port-specific record does not exist. The scanning engine will authenticate to the database instance on each port the Oracle service is detected on. You may only create one Oracle authentication record with this setting for each host.

A single port is used when the **port** parameter is specified (e.g. **port=1521**). The same port number cannot be entered in multiple Oracle records for the same host, unless the compliance module is enabled and **pc_only=1** is specified.

When the scanning engine detects an Oracle instance on a host, it first checks to see if you have an authentication record with the database instance and port specified. If you have a port-specific record, then it uses the credentials in that record to attempt authentication to the database instance. If a port-specific record does not exist (or if authentication fails), then the scanning engine checks to see if you have an authentication record set to "All Ports" for the host and uses the credentials in that record to attempt authentication to the database instance.

Oracle Use Cases

See the topic "Oracle Use Cases" in the Qualys online help to learn more about how to define Oracle authentication records for various network configurations.

Oracle Record Properties: Target Hosts

The same IP may be included in multiple Oracle records as long as different ports are specified. Each IP may be included in one record with the "All Ports" setting.

The following parameters are used to define the target hosts that the scanning engine should log into with the specified credentials.

Parameter	Description
ips={value}	(Required for create request) Defines an IP list for the authentication record. (Optional for update request) Overwrites (replaces) the IP list for the authentication record. The IPs you specify are added and any existing IPs are removed. You may enter a combination of IPs and IP ranges. Multiple entries are comma separated.
	This parameter and the **add_ips** parameter or the **remove_ips** parameter cannot be specified in the same request.
add_ips={value}	(Optional for update request only) This parameter is used to update an existing IP list an existing authentication record. Specifies one or more IP addresses to add to the IP list for the authentication record. You may enter a combination of IPs and IP ranges. Multiple entries are comma separated.
	This parameter and the **ips** parameter cannot be specified in the same request.
remove_ips={value}	(Optional for update request only) This parameter is used to update an existing IP list in an existing authentication record. Specifies one or more IP addresses to remove from the IP list for the authentication record. You may enter a combination of IPs and IP ranges. Multiple entries are comma separated.
	This parameter and the **ips** parameter cannot be specified in the same request.

Oracle Record Properties: OS Parameters

OS parameters are used by compliance scans only.

Windows Parameters

Define Windows parameters to be used for Windows compliance scans.

Parameter	Description
perform_windows_os_checks={**0** | 1}	(Optional) Specify 1 to perform OS-dependent compliance checks for the Oracle technology during Windows authenticated compliance scans. These checks are assigned to the control category "Databse Setttings" in the sub-category "DB OS-dependent Controls".
win_ora_home_name={value}	(Required if perform_windows_os_checks=1 is specified; otherwise invalid) The Windows Oracle Home name. Example: OraHome1
win_ora_home_path={value}	(Required if perform_windows_os_checks=1 is specified; otherwise invalid) The Windows Oracle Home path. Example: c:\Program Files\Oracle\10
win_init_ora_path={value}	(Required if perform_windows_os_checks=1 is specified; otherwise invalid) The pathname to the Windows init(SID).ora file. Example: c:\Program Files\oracle\dbs\initORA10.ora
win_spfile_ora_path={value}	(Required if perform_windows_os_checks=1 is specified; otherwise invalid) The pathname to the Windows spfile(SID).ora file. Example: c:\Program Files\oracle\network\admin\spfileORA10.ora
win_listener_ora_path={value}	(Required if perform_windows_os_checks=1 is specified; otherwise invalid) The pathname to the Window listener.ora file. Example: c:\Program Files\oracle\network\admin\listener.ora
win_sqlnet_ora_path={value}	(Required if perform_windows_os_checks=1 is specified; otherwise invalid) The pathname to the Windows sqlnet.ora file. Example: c:\Program Files\oracle\network\admin\sqlnet.ora
win_tnsnames_ora_path={value}	(Required if perform_windows_os_checks=1 is specified; otherwise invalid) The pathname to the Windows tnsnames.ora file. Example: c:\ProgramFiles\oracle\network\admin\tnsnames.ora

Unix Parameters

Define Unix parameters to be used for Unix compliance scans.

Parameter	Description
perform_unix_os_checks={**0**|1}	(Optional) Specify 1 to perform OS-dependent compliance checks for the Oracle technology during Unix authenticated compliance scans. These checks are assigned to the control category "Databse Setttings" in the sub-category "DB OS-dependent Controls".
perform_unix_opatch_checks={**0**|1}	(Optional) Specify 1 to perform OPatch checks using the OPatch binary to return a list of all installed patches for the Oracle instance. In a case where perform_unix_os_checks=1 is specified and perform_unix_opatch_checks=0 is specified (or this parameter is not specified), the service checks for patch information from the Oracle database directly; information in the database may not be accurate so the list of installed patches returned by the service also may not be accurate.
unix_ora_home_path={value}	(Required if perform_unix_os_checks=1 and/or perform_unix_opatch_checks=1 is specified; otherwise invalid) The Unix Oracle Home path. Example: /usr/opt/oracle/10
unix_init_ora_path={value}	(Required if perform_unix_os_checks=1 and/or perform_unix_opatch_checks=1 is specified; otherwise invalid) The pathname to the Unix init(SID).ora file. Example: /usr/opt/oracle/dbs/initORA10.ora
unix_spfile_ora_path={value}	(Required if perform_unix_os_checks=1 and/or perform_unix_opatch_checks=1 is specified; otherwise invalid) The pathname to the Unix spfile(SID).ora file. Example: /usr/opt/oracle/network/admin/spfileORA10.ora
unix_listener_ora_path={value}	(Required if perform_unix_os_checks=1 and/or perform_unix_opatch_checks=1 is specified; otherwise invalid) The pathname to the Unix listener.ora file. Example: /usr/opt/oracle/network/admin/listener.ora
unix_sqlnet_ora_path={value}	(Required if perform_unix_os_checks=1 and/or perform_unix_opatch_checks=1 is specified; otherwise invalid) The pathname to the Unix sqlnet.ora file. Example: /usr/opt/oracle/network/admin/sqlnet.ora

Parameter	Description
unix_tnsnames_ora_path={value}	(Required if perform_unix_os_checks=1 and/or perform_unix_opatch_checks=1 is specified; otherwise invalid) The pathname to the Unix tnsnames.ora file. Example: /usr/opt/oracle/network/admin/tnsnames.ora
unix_invptrloc={value}	(Optional) if perform_unix_opatch_checks=1 is specified; otherwise invalid) The pathname to the Unix oraInst.loc file. Use this parameter to identify a custom inventory for patches. Example: /usr/opt/oracle/network/admin/oraInst.loc

Oracle Record: XML Output

The XML output returned by an Oracle record API request (create, edit or delete) uses the DTD at the following URL (where qualysapi.qualys.com is the API server URL where your account is located):

```
https://qualysapi.qualys.com/api/2.0/batch_return.dtd
```

Want to learn more? Please see Appendix D.

Oracle Listener Record

The **/api/2.0/fo/auth/oracle_listener** resource allows you to manage Oracle Listener authentication records for authenticated scanning. You can submit API requests to view Oracle Listener authentication records, add new records, update records and delete records. Authentication is required for each API request. See Chapter 2, "Authentication Using the V2 APIs."

Oracle Listener records are used to connect to Oracle TNS Listeners in order to enumerate information about databases behind the Oracle Listeners. When authentication is successful and databases behind the Listener are discovered, the QID 19225 "Retrieved Oracle Database Name" is returned in the vulnerability scan results. This is an information gathered check that lists the names of the databases discovered behind the Listener. This information is useful if you want to create Oracle authentication records on those databases and need the Oracle System IDs (SIDs).

Multiple Oracle Listener records with different passwords may be created for each host.

Maximum Authentication Records per API Request

A maximum of 1,000 authentication records can be processed per request, when requesting records to view, update or delete in batch mode. If the requested list identifies more than 1,000 authentication records, then the XML output includes the <WARNING> element and instructions for making another request for the next batch of records.

User Permissions

See "User Permissions and Authentication Records" for information.

Oracle Listener Record: Request

URL. Use this URL to make an Oracle Listener authentication record request (create, update, or delete), <qualysapi.qualys.com> is the API server URL where your account is located. Replace the API server URL if your account is located on another platform.

```
https://<qualysapi.qualys.com>/api/2.0/fo/auth/oracle_listener
```

Method. For a list request, the GET or POST method may be used. For another request (create, update, or delete), the POST access method must be used.

Authentication. Authentication is required for each API request. See Chapter 2, "Authentication Using the V2 APIs."

Oracle Listener Record: List

Specify a request that returns a list of Oracle Listener authentication records. See "Authentication Record List by Type" for details.

Oracle Listener Record: Create

The parameters described below are used to specify a request that creates a new Oracle Listener authentication record.

Parameter	Description
action=create	(Required) Specifies the create action type for the request.
echo_request={**0** | 1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
title={value}	(Required) Specifies a title for the authentication record. The title must be unique and may include a maximum of 255 characters (ascii).
comments={value}	(Optional) Specifies user defined notes about the authentication record. The comments may include a maximum of 1999 characters (ascii); if comments have 2000 or more characters an error is returned and comments are not saved. Tags (such as <script>) cannot be included; if tags are included an error is returned and the request fails.
password={value}	(Required) Specifies a password for authentication to target hosts. If more than one Listener is detected on the same host, then the same password is attempted on each Listener. The password may include 1-31 characters (ascii).
{target hosts}	Define the target hosts for authentication. See "Oracle Listener Record Properties: Target Hosts."
network_id={value}	(Optional and valid when the networks feature is enabled) The network ID for the record.

Oracle Listener Record: Update

The parameters described below are used to specify a request to update one or more existing Oracle Listener authentication records. Parameters specified in an update request will overwrite any existing parameters (previously defined).

Parameter	Description
action=update	(Required) Specifies the update action type for the request.
echo_request={**0**|1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
ids={value}	(Required) Update only authentication records with certain IDs and/or ID ranges. Multiple entries are comma separated. One or more IDs/ranges may be specified. An ID range entry is specified with a hyphen (for example, 1359-1407). Valid IDs are required.
title={value}	(Optional) Overwrites the existing record title with a new title. The title must be unique and may include a maximum of 255 characters (ascii). When multiple IDs are specified for a batch update, the title is only replaced in the first record ID in the action since the title must be unique across records. The update will fail for the remaining IDs included in the action.
comments={value}	(Optional) Specifies user defined notes about the authentication record. The comments may include a maximum of 1999 characters (ascii); if comments have 2000 or more characters an error is returned and comments are not saved. Tags (such as <script>) cannot be included; if tags are included an error is returned and the request fails.
password={value}	(Optional) Specifies a password for authentication to target hosts. If more than one Listener is detected on the same host, then the same password is attempted on each Listener. The password may include 1-31 characters (ascii).
{target hosts}	Define the target hosts for authentication. See "Oracle Listener Record Properties: Target Hosts."
network_id={value}	(Optional and valid when the networks feature is enabled) The network ID for the record.

Oracle Listener Record: Delete

The parameters described below are used to specify a request that deletes one or more Oracle Listener authentication records.

Parameter	Description
action=delete	(Required) Specifies the delete action type for the request.
echo_request={**0**|1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
ids={value}	(Required) Delete only authentication records with certain IDs and/or ID ranges. Multiple entries are comma separated. One or more IDs/ranges may be specified. An ID range entry is specified with a hyphen (for example, 3000-3250). Valid IDs are required.

Oracle Listener Record Properties: Target Hosts

Each IP address in your account may be included in one Oracle Listener record.

The following parameters are used to define the target hosts that the scanning engine should log into with the specified credentials.

Parameter	Description
ips={value}	(Required for create request) Defines an IP list for the authentication record. (Optional for update request) Overwrites (replaces) the IP list for the authentication record. The IPs you specify are added and any existing IPs are removed. You may enter a combination of IPs and IP ranges. Multiple entries are comma separated. This parameter and the **add_ips** parameter or the **remove_ips** parameter cannot be specified in the same request.

Parameter	Description
add_ips={value}	(Optional for update request only) This parameter is used to update an existing IP list an existing authentication record. Specifies one or more IP addresses to add to the IP list for the authentication record. You may enter a combination of IPs and IP ranges. Multiple entries are comma separated. This parameter and the **ips** parameter cannot be specified in the same request.
remove_ips={value}	(Optional for update request only) This parameter is used to update an existing IP list in an existing authentication record. Specifies one or more IP addresses to remove from the IP list for the authentication record. You may enter a combination of IPs and IP ranges. Multiple entries are comma separated. This parameter and the **ips** parameter cannot be specified in the same request.

Oracle Listener Record: XML Output

The XML output returned by an Oracle Listener record API request (create, edit or delete) uses the DTD at the following URL (where qualysapi.qualys.com is the API server URL where your account is located):

```
https://qualysapi.qualys.com/api/2.0/batch_return.dtd
```

Want to learn more? Please see Appendix D.

SNMP Record

The **/api/2.0/fo/auth/snmp** resource allows you to manage SNMP authentication records for authenticated scanning. Authentication is required for each API request. See Chapter 2, "Authentication Using the V2 APIs."

Using this API resource you can submit API requests to view SNMP authentication records, add new records, update records and delete records. Authenticated scanning using these SNMP versions is supported: SNMPv1, SNMPv2 and SNMPv3.

Important: For vulnerability scans, privileged admin access is optional; when provided the service has the ability to perform more in depth security analysis. For compliance scans, privileged admin access is required.

Maximum Authentication Records per API Request

A maximum of 1,000 authentication records can be processed per request, when requesting records to view, update or delete in batch mode. If the requested list identifies more than 1,000 authentication records, then the XML output includes the <WARNING> element and instructions for making another request for the next batch of records.

User Permissions

See "User Permissions and Authentication Records" for information.

SNMP Record: Request

URL. Use this URL to make an SNMP authentication record request (create, update, or delete), where <qualysapi.qualys.com> is the API server URL where your account is located. Replace the API server URL if your account is located on another platform.

```
https://<qualysapi.qualys.com>/api/2.0/fo/auth/snmp
```

Method. For a list request, the GET or POST method may be used. For another request (create, update, or delete), the POST access method must be used.

Authentication. Authentication is required for each API request. See Chapter 2, "Authentication Using the V2 APIs."

SNMP Record: List

Specify a request that returns a list of SNMP authentication records. See "Authentication Record List by Type" for details.

SNMP Record: Create

The parameters described below are used to specify a request that creates a new SNMP authentication record.

Parameter	Description
action=create	(Required) Specifies the create action type for the request.
echo_request={**0** | 1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
title={value}	(Required) Specifies a title for the authentication record. The title must be unique and may include a maximum of 255 characters (ascii).
comments={value}	(Optional) Specifies user defined notes about the authentication record. The comments may include a maximum of 1999 characters (ascii); if comments have 2000 or more characters an error is returned and comments are not saved. Tags (such as <script>) cannot be included; if tags are included an error is returned and the request fails.
version={**v1** | v2c | v3}	(Optional) Specifies the SNMP protocol version. A valid value is: v1 = SNMPv1 (the default) v2c = SNMPv2c v3 = SNMPv3
{login credentials}	Define login credentials for authentication to target hosts. See "SNMP Record Properties: Login Credentials."
{target hosts}	Define the target hosts for authentication. See "SNMP Record Properties: Target Hosts."
network_id={value}	(Optional and valid when the networks feature is enabled) The network ID for the record.

SNMP Record: Update

The parameters described below are used to specify a request to update one or more existing SNMP authentication records. Parameters specified in an update request will overwrite any existing parameters (previously defined).

Each IP address in your account may be included in one SNMP record.

Parameter	Description
action=update	(Required) Specifies the update action type for the request.
echo_request={**0** | 1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
ids={value}	(Required) Update only authentication records with certain IDs and/or ID ranges. Multiple entries are comma separated. One or more IDs/ranges may be specified. An ID range entry is specified with a hyphen (for example, 1359-1407). Valid IDs are required.
title={value}	(Optional) Overwrites the existing record title with a new title. The title must be unique and may include a maximum of 255 characters (ascii). When multiple IDs are specified for a batch update, the title is only replaced in the first record ID in the action since the title must be unique across records. The update will fail for the remaining IDs included in the action.
comments={value}	(Optional) Specifies user defined notes about the authentication record. The comments may include a maximum of 1999 characters (ascii); if comments have 2000 or more characters an error is returned and comments are not saved. Tags (such as <script>) cannot be included; if tags are included an error is returned and the request fails.
version={v1 | v2c | v3}	(Optional) Overwrites the existing SNMP version with a new version. A valid value is: v1 = SNMPv1 v2c = SNMPv2c v3 = SNMPv3
{login credentials}	Define login credentials for authentication to target hosts. See "SNMP Record Properties: Login Credentials."

Parameter	Description
{target hosts}	Define the target hosts for authentication. See "SNMP Record Properties: Target Hosts."
network_id={value}	(Optional and valid when the networks feature is enabled) The network ID for the record.

SNMP Record: Delete

The parameters described below are used to specify a request that deletes one or more SNMP authentication records.

Parameter	Description
action=delete	(Required) Specifies the delete action type for the request.
echo_request={**0**|1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
ids={value}	(Required) Delete only authentication records with certain IDs and/or ID ranges. Multiple entries are comma separated. One or more IDs/ranges may be specified. An ID range entry is specified with a hyphen (for example, 3000-3250). Valid IDs are required.

SNMP Record Properties: Login Credentials

The **community_strings** parameter may be used for SNMPv1 and SNMPv2c. This parameter is not valid for records set to SNMPv3.

Parameter	Description
community_strings={value}	(Optional) Specifies the SNMP community strings to be used for authentication to target hosts. The service attempts authentication using several common default community strings. When **community_strings** is specified, the user-provided community strings are used for authentication before the default community strings. Multiple entries are comma separated.

The following parameters may be used for SNMPv3.

Parameter	Description
username={value}	(Optional) Specifies the user account for authentication to target hosts. A maximum of 128 characters may be specified.
	These three parameters are used to specify authentication: **`username`**, **`password`** and **`auth_alg`**. If creating a record and authentication will be used, it is required that all three parameters are specified together. If updating a record to change the username, the username specified will replace the existing username in the record. If updating a record to remove authentication, specify an empty value for all three parameters.
password={value}	(Optional) Specifies a password for authentication to target hosts. A maximum of 128 characters may be specified.
	These three parameters are used to specify authentication: **`username`**, **`password`** and **`auth_alg`**. If creating a record and authentication will be used, it is required that all three parameters are specified together. If updating a record to change the password, the password specified will replace the existing password in the record. If updating a record to remove authentication, specify an empty value for all three parameters.
auth_alg={MD5|SHA1}	(Optional) Specifies the algorithm for authentication: MD5 or SHA1. This algorithm is used to safely prove to the SNMP server knowledge of the password without sending the password.
	These three parameters are used to specify authentication: **`username`**, **`password`** and **`auth_alg`**. If creating a record and authentication will be used, it is required that all three parameters are specified together. If updating a record to change the authentication algorithm, the algorithm specified will replace the existing algorithm in the record. If updating a record to remove authentication, specify an empty value for all three parameters.

Parameter	Description
encrypt_password={value}	(Optional) Specifies the password if privacy (data encryption) is to be used for SNMP communication. A maximum of 128 characters may be specified.
	These two parameters are used to specify privacy: **`encrypt_password`** and **`priv_alg`**. If creating a record and privacy will be used, it is required that both parameters are specified together. If updating a record to change the password, the password specified will replace the existing password in the record. If updating a record to remove privacy, specify an empty value for both parameters.
priv_alg={DES | AES}	(Optional) Specifies the algorithm to be used for privacy: DES or AES. This algorithm is used to encrypt and decrypt SNMP messages.
	These two parameters are used to specify privacy: **`encrypt_password`** and **`priv_alg`**. If creating a record and privacy will be used, it is required that both parameters are specified together. If updating a record to change the privacy algorithm, the algorithm specified will replace the existing algorithm in the record. If updating a record to remove privacy, specify an empty value for both parameters.
security_engine_id={value}	(Optional) Specifies the security engine ID when a security engine is part of the target host configuration. A valid ID is required. A maximum of 128 characters may be specified.
	If a security engine ID is part of the target host configuration, the parameter **`security_engine_id`** must be defined for the record in order for authentication to be successful. If the security engine ID is not defined (and is required by the target host for all SNMP requests), then the SNMP service may not be detected on the target host and authentication will fail.
context_engine_id={value}	(Optional) Specifies the context engine ID used in scoped PDUs when a context is part of the target host configuration. A valid ID is required. A maximum of 128 characters may be specified.
	If an SNMP context is part of the target host configuration, the parameters **`context_engine_id`** and/or **`context`** must be defined for the record in order for the scanning engine to retrieve context-sensitive information from the target host.

Parameter	Description
context={value}	(Optional) Specifies the context name used in scoped PDUs when a context is part of the target host configuration. A maximum of 128 characters may be specified.
	If an SNMP context is part of the target host configuration, the parameters **`context_engine_id`** and/or **`context`** must be defined for the record in order for the scanning engine to retrieve context-sensitive information from the target host.

SNMP Record Properties: Target Hosts

Each IP address in your account may be included in one SNMP record.

The following parameters are used to define the target hosts that the scanning engine should log into with the specified credentials.

Parameter	Description
ips={value}	(Required for create request) Defines an IP list for the authentication record. (Optional for update request) Overwrites (replaces) the IP list for the authentication record. The IPs you specify are added and any existing IPs are removed. You may enter a combination of IPs and IP ranges. Multiple entries are comma separated.
	This parameter and the **`add_ips`** parameter or the **`remove_ips`** parameter cannot be specified in the same request.
add_ips={value}	(Optional for update request only) This parameter is used to update an existing IP list an existing authentication record. Specifies one or more IP addresses to add to the IP list for the authentication record. You may enter a combination of IPs and IP ranges. Multiple entries are comma separated.
	This parameter and the **`ips`** parameter cannot be specified in the same request.

Parameter	Description
remove_ips={value}	(Optional for update request only) This parameter is used to update an existing IP list in an existing authentication record. Specifies one or more IP addresses to remove from the IP list for the authentication record. You may enter a combination of IPs and IP ranges. Multiple entries are comma separated. This parameter and the **ips** parameter cannot be specified in the same request.

SNMP Record: Sample API Requests

These sample API requests work on Qualys US Platform 1 where the FQDN in the API server URL is qualysapi.qualys.com. Please be sure to replace the FQDN with the proper API server URL for your platform. For the EU platform, use qualysapi.qualys.eu. For a partner platform, use the URL for your @customer platform API server.

The header parameter "X-Requested-With" is also provided as an example. Please change this parameter according to your need. This parameter is exposed in the activity log (only displayed when used with session cookie authentication - does not apply for basic authentication as used in the following examples) and it is useful for API monitoring API activity.

Sample 1. Use the URL shown below to create a new record for SNMPv3:

```
curl -H 'X-Requested-With: Curl Sample' -d
'action=create&title=My+Record&version=v3&username=user&password=pass
word&auth_alg=MD5&encrypt_password=passwordabcde123456&priv_alg=DES&s
ecurity_engine_id=0x80001F88805131F121BD9B194B&context_engine_id=0x80
001F88805131F121BD9B194B&context=bridge1&ips=10.10.10.2-10.10.10.4' -
b 'QualysSession=a3863e31b486417f81eea7f8881f3142; path=/api; secure'
'https://qualysapi.qualys.com/api/2.0/fo/auth/snmp/'
```

Sample 2. Use the URL shown below to update an existing SNMP record to change the user name and password for authentication and the IPs in the record:

```
curl  -H 'X-Requested-With: Curl Sample' -d
'action=update&ids=65319&username=user2&password=password2&ips=10.10.
10.5-10.10.10.6'
-b 'QualysSession=a3863e31b486417f81eea7f8881f3142; path=/api;
secure' 'https://qualysapi.qualys.com/api/2.0/fo/auth/snmp/'
```

SNMP Record: XML Output

The XML output returned by an SNMP record API request (create, edit or delete) uses the DTD at the following URL (where qualysapi.qualys.com is the API server URL where your account is located):

```
https://qualysapi.qualys.com/api/2.0/batch_return.dtd
```

Want to learn more? Please see Appendix D.

IBM DB2 Record

The **/api/2.0/fo/auth/ibm_db2/** resource allows you to manage DB2 authentication records. You can submit API requests to view DB2 authentication records, add new records, update records and delete records. Authentication is required for each API request. See Chapter 2, "Authentication Using the V2 APIs."

During scanning the service will authenticate to one or more DB2 instances on a single host using all the DB2 records in your account. See "Multiple DB2 Instances."

Maximum Authentication Records per API Request

A maximum of 1,000 authentication records can be processed per request, when requesting records to view, update or delete in batch mode. If the requested list identifies more than 1,000 authentication records, then the XML output includes the <WARNING> element and instructions for making another request for the next batch of records.

User Permissions

See "User Permissions and Authentication Records" for information.

DB2 Record: Request

URL. Use this URL to make an DB2 authentication record request (create, update, or delete), where <qualysapi.qualys.com> is the API server URL where your account is located. Replace the API server URL if your account is located on another platform.

```
https://<qualysapi.qualys.com>/api/2.0/fo/auth/ibm_db2
```

Method. For a list request, the GET or POST method may be used. For another request (create, update, or delete), the POST access method must be used.

Authentication. Authentication is required for each API request. See Chapter 2, "Authentication Using the V2 APIs."

DB2 Record: List

Specify a request that returns a list of DB2 authentication records. See "Authentication Record List by Type" for details.

DB2 Record: Create

The parameters described below are used to specify a request that creates a new DB2 authentication record.

Parameter	Description
action=create	(Required) The action type for the API request: create.
echo_request={**0** | 1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When unspecified, parameters are not included in the XML output.
title={value}	(Required) The title for the record. The title must be unique and may include a maximum of 255 characters (ascii).
comments={value}	(Optional) User defined notes about the record. The comments may include a maximum of 1999 characters (ascii); if comments have 2000 or more characters an error is returned and comments are not saved. Tags (such as <script>) cannot be included; if tags are included an error is returned and the request fails.
{login credentials}	(Required) See "DB2 Record Properties: Login Credentials."
ips={value}	(Required) Add IP address(es) to the IP list for a new authentication record. You may enter a combination of IPs and IP ranges. Multiple entries are comma separated.
pc_only={**0** | 1}	(Optional) Specify pc_only=1 if the record will be used for compliance scans only. See "Multiple DB2 Instances."
network_id={value}	(Optional and valid when the networks feature is enabled) The network ID for the record.

DB2 Record: Update

The parameters listed below are used to specify a request to update one or more existing DB2 authentication records. Parameters specified in an update request will overwrite any existing parameters (previously defined).

Parameter	Description
action=update	(Required) The action type for the API request: update.
echo_request={**0** | 1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When unspecified, parameters are not included in the XML output.

Parameter	Description
ids={value}	(Required) Update only authentication records with certain IDs and/or ID ranges. Multiple entries are comma separated. One or more IDs/ranges may be specified. An ID range entry is specified with a hyphen (for example, 1359-1407). Valid IDs are required.
title={value}	(Optional) Overwrites the existing record title with a new title. The title must be unique and may include a maximum of 255 characters (ascii). When multiple IDs are specified for a batch update, the title is only replaced in the first record ID in the action since the title must be unique across records. The update will fail for the remaining IDs included in the action.
comments={value}	(Optional) User defined notes about the record. A maximum of 1999 characters (ascii) may be specified.
{login credentials}	(Optional) See "DB2 Record Properties: Login Credentials."
ips={value}	Overwrites (replaces) the IP address(es) in the IP list for an existing authentication record. The IPs you specify are added, and any existing IPs are removed. You may enter a combination of IPs and IP ranges. Multiple entries are comma separated. See "Multiple DB2 Instances" below.
add_ips={value}	(Optional) Add IP address(es) to the IP list for an existing authentication record. You may enter a combination of IPs and IP ranges. Multiple entries are comma separated. See "Multiple DB2 Instances." below.
remove_ips={value}	(Optional) Remove IP address(es) from the IP list for an existing authentication record. You may enter a combination of IPs and IP ranges. Multiple entries are comma separated.
pc_only={**0** | 1}	(Optional) Specify pc_only=1 if the record will be used for compliance scans only. See "Multiple DB2 Instances."
network_id={value}	(Optional and valid when the networks feature is enabled) The network ID for the record.

DB2 Record: Delete

The parameters listed below are used to specify a request that deletes one or more DB2 authentication records.

Parameter	Description
action=delete	(Required) The action required for the API request: delete.
echo_request={**0** | 1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output.
ids={value}	(Required) Delete only authentication records with certain IDs and/or ID ranges. Multiple entries are comma separated. One or more IDs/ranges may be specified. An ID range entry is specified with a hyphen (for example, 3000-3250). Valid IDs are required.

DB2 Record Properties: Login Credentials

The following parameters are used to define the login credentials. Parameters specified in an update request are optional and will overwrite any existing parameters (previously defined).

Parameter	Description
username={value}	(Required for a create request; optional for an update request) The user name for a DB2 database account. A maximum of 13 characters (ascii) may be specified.
password={value}	(Required for a create request; optional for an update request) The password for a DB2 database account. A maximum of 13 characters (ascii) may be specified.
database={value}	(Required for a create request; optional for an update request) The name of the DB2 database. A maximum of 8 characters (ascii) may be specified. See "Multiple DB2 Instances" below.
port={value}	(Required for a create request; optional for an update request) The port the database instance is running on. See "Multiple DB2 Instances"below.

Multiple DB2 Instances

The service has the ability to authenticate to multiple DB2 instances on a single host during scanning. For a vulnerability scan, an instance "uniqueness" is defined by an IP address and port. For a compliance scan, an instance "uniqueness" is defined by an IP address, port and database name. The setting for "pc_only" has an impact on how the services determines the uniqueness of a DB2 instance.

Let's say you want to define these DB2 records in your account.

	IP Address	Port	Database Name	pc_only=0|1
Record 1	10.10.31.178	5000	SAMPLE	pc_only=0
Record 2	10.10.30.159	5000	TOOLS	pc_only=0
Record 3	10.10.30.159	5000	SAMPLE	pc_only=1

Record 1 and Record 2 will be used for vulnerability scans and compliance scans. You'll notice Records 2 and 3 have the same IP address and port but different database names - this is allowed because Record 3 is used for compliance scans only.

DB2 Record: Windows and Unix Parameters

The parameters described below are used to define the Windows and/or Unix record(s) in your account for your DB2 hosts if you want the service to perform OS-dependent compliance checks. Provide all the following Windows parameters if you want the service to gather DB2 compliance data from Windows hosts. Provide all the following Unix parameters if you want the service to gather DB2 compliance data from Unix hosts.

Parameter	Description
win_db2dir={value} unix_db2dir={value}	The path to the DB2 runtime library if you want the service to perform OS-dependent compliance checks. This is the location where DB2 has been installed on the server. A maximum of 255 characters may be specified. See "DB2 Paths" below.
win_prilogfile={value} unix_prilogfile={value}	The path to the primary archive location if you want the service to perform OS-dependent compliance checks. This is the directory where the primary log files are located. A maximum of 255 characters may be specified. See "DB2 Paths" below.
win_seclogfile={value} unix_seclogfile={value}	The path to the secondary archive location if you want the service to perform OS-dependent compliance checks. This parameter specifies the number of secondary log files that are created and used for recovery log files (only as needed). It is set by the DB2 logsecond parameter. A maximum of 255 characters may be specified. See "DB2 Paths" below.

Parameter	Description
win_terlogfile={value} unix_terlogfile={value}	The path to the tertiary archive location if you want the service to perform OS-dependent compliance checks. This parameter specifies a path to which DB2 will try to archive log files if the log files cannot be archived to either the primary or the secondary (if set) archive destinations because of a media problem affecting those destinations. It is set by the DB2 failarchpath parameter. A maximum of 255 characters may be specified. See "DB2 Paths" below.
win_mirlogfile={value} unix_mirlogfile={value}	The path to the mirror archive location if you want the service to perform OS-dependent compliance checks. If mirrorlogpath is configured, DB2 will create active log files in both the log path and the mirror log path. All log data will be written to both paths. The mirror log path has a duplicate set of active log files. If the active log files are destroyed by a disk error or human error, the database can still function. A maximum of 255 characters may be specified. See "DB2 Paths" below.

DB2 Paths

When specifying the path to configuration files, these special characters are not allowed:

For Windows:

; & | # % ? ! * ` ( ) [ ] " ' > < = ^ /

For Unix:

; & | # % ? ! * ` ( ) [ ] " ' > < = ^ \

IBM DB2 Record: XML Output

The XML output returned by a DB2 record API request (create, edit or delete) uses the DTD at the following URL)where qualysapi.qualys.com is the API server URL where your account is located_:

```
https://qualysapi.qualys.com/api/2.0/batch_return.dtd
```

Want to learn more? Please see Appendix D.

VMware Record

The VMware authentication record type allows for connections to the vSphere API for vSphere 5.x and 4.x. The vSphere API is a SOAP API used by all vSphere components, including VMware ESXi, VMware ESX, VMware vCenter Server, and the VMware vCenter Server Appliance. By default, the API connection occurs over an encrypted SSL web services connection on port 443. The existence of this authentication module immediately enables several Qualys capabilities for Mapping, Vulnerability Management, and Policy Compliance for vSphere; and will support more vSphere-focused capabilities and integrations in the future.

The **`/api/2.0/fo/auth/vmware/`** resource allows you to manage VMware authentication records. You can submit API requests to view VMware authentication records, add new records, update records and delete records. Authentication is required for each API request. See Chapter 2, "Authentication Using the V2 APIs."

Maximum Authentication Records per API Request

A maximum of 1,000 authentication records can be processed per request, when requesting records to view, update or delete in batch mode. If the requested list identifies more than 1,000 authentication records, then the XML output includes the <WARNING> element and instructions for making another request for the next batch of records.

User Permissions

See "User Permissions and Authentication Records" for information.

Input Parameters

The input parameters for the VMware authentication record API are described below.

VMware Record: Create

These parameters are used to create a VMware authentication record.

Parameter	Description
action=create	(Required) The action type for the API request: create.
echo_request={**0**|1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When unspecified, parameters are not included in the XML output.
title={value}	(Required) The title of the VMware record. The title must be unique and may include a maximum of 255 characters (ascii).

Parameter	Description
comments={value}	(Optional) User defined notes about the VMware record. The comments may include a maximum of 1999 characters (ascii); if comments have 2000 or more characters an error is returned and comments are not saved. Tags (such as <script>) cannot be included; if tags are included an error is returned and the request fails.
{login credentials}	(Required) See "VMware Record: Login Credentials."
ips={value}	(Required) Add IP address(es) of the ESXi servers that the scanning engine should log into using the record's credentials. You may enter a combination of IPs and IP ranges. Multiple entries are comma separated.
network_id={value}	(Optional and valid when the networks feature is enabled) The network ID for the record.

VMware Record: Update

These parameters are used to update an existing VMware authentication record.

Parameter	Description
action=update	(Required) The action type for the API request: update.
echo_request={**0** | 1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When unspecified, parameters are not included in the XML output.
ids={value}	(Required) Update only VMware records with certain IDs and/or ID ranges. Multiple entries are comma separated. One or more IDs/ranges may be specified. An ID range entry is specified with a hyphen (for example, 1359-1407). Valid IDs are required.
title={value}	(Optional) Overwrites the existing VMware record title with a new title. The title must be unique and may include a maximum of 255 characters (ascii). When multiple IDs are specified for a batch update, the title is only replaced in the first record ID in the action since the title must be unique across records. The update will fail for the remaining IDs included in the action.
comments={value}	(Optional) User defined notes about the VMware record. A maximum of 1999 characters (ascii) may be specified.
{login credentials}	(Optional) See "VMware Record: Login Credentials."

Parameter	Description
ips={value}	(Optional) Overwrites (replaces) the IP address(es) in the IP list for an existing VMware record. The IPs you specify are added, and any existing IPs are removed. You may enter a combination of IPs and IP ranges. Multiple entries are comma separated.
add_ips={value}	(Optional) Add IP address(es) to the IP list for an existing VMware record. You may enter a combination of IPs and IP ranges. Multiple entries are comma separated.
remove_ips={value}	(Optional) Remove IP address(es) from the IP list for an existing VMware record. You may enter a combination of IPs and IP ranges. Multiple entries are comma separated.
network_id={value}	(Optional and valid when the networks feature is enabled) The network ID for the record.

VMware Record: Delete

These parameters are used to delete an existing VMware authentication record.

Parameter	Description
action=delete	(Required) The action required for the API request: delete.
echo_request={**0**|1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output.
ids={value}	(Required) Delete only VMware records with certain IDs and/or ID ranges. Multiple entries are comma separated. One or more IDs/ranges may be specified. An ID range entry is specified with a hyphen (for example, 3000-3250). Valid IDs are required.

VMware Record: Login Credentials

The login credentials for authenticating to a VMware ESXi server are described below.

Parameter	Description
username={value}	(Required for a create request; optional for an update request) The user name for a VMware account. A maximum of 13 characters (ascii) may be specified.
password={value}	(Required for a create request; optional for an update request) The password for a VMware account. A maximum of 13 characters (ascii) may be specified.
port={value}	(Optional) The service communicates with ESXi web services on port 443 and another port can be configured. When unspecified, port 443 is used.

Parameter	Description
hosts={value}	(Optional) A list of FQDNs for the hosts that correspond to all ESXi host IP addresses on which a custom SSL certificate signed by a trusted root CA is installed. Multiple hosts are comma separated.
ssl_verify={value}	(Optional) Specify "all" for a complete SSL certificate validation. Specify "skip" if the host SSL certificate is self-signed or uses an SSL certificate signed by a custom root CA. Specify "none" for no SSL verification.

VMware Record: XML Output

The XML output returned by a VMware record API request (create, edit or delete) uses the DTD at the following URL (where qualysapi.qualys.com is the API server URL where your account is located):

```
https://qualysapi.qualys.com/api/2.0/batch_return.dtd
```

Want to learn more? Please see Appendix D.

VMware Record: Sample API Request

API Request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d
"action=create&title=NewVMwareRecordWithAPI&username=USERNAME&passwor
d=PASSWORD&ips=10.10.10.2-10.10.10.4"
"https://qualysapi.qualys.com/api/2.0/fo/auth/vmware/" >
apiOutputCreateVMwareRecord.txt
```

XML Output:

```
cat apiOutputCreateVMwareRecord.txt

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/auth/vmware/batch_return.dtd
">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2012-02-03T21:16:41Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>30486</ID>
```

```
            </ID_SET>
          </BATCH>
        </BATCH_LIST>
      </RESPONSE>
    </BATCH_RETURN>
```

Application Server Records

Policy Compliance includes content in the control Knowledgebase to provide support of Application Server Technologies. You can create and manage these authentication records using the "Authentication" API v2 (resource **`/api/2.0/fo/auth/`**):

Application Server Technology Record	Authentication API v2 endpoint
Apache 2.2 - IBM HTTP Server 7.x (on Red Hat Linux 5/6) - VMware vFabric Web Server 5.2 (on Red Hat Linux 5/6)	/api/2.0/fo/auth/apache/
Microsoft Internet Information Services (IIS) 6.0, 7.x	/api/2.0/fo/auth/ms_iis/
IBM WebSphere Application Server 7.0	/api/2.0/fo/auth/ibm_websphere/
Tomcat Server - Apache Tomcat 6.x and 7.x - VMware vFabric tc Server 2.9.x - Pivotal tc Server 3.x	/api/2.0/fo/auth/tomcat/

Using the Authentication Server API you can submit API requests to view Server authentication records, add new records, update records and delete records. Authentication is required for each API request. See Chapter 2, "Authentication Using the V2 APIs."

Apache server authentication - support for multiple instances per host. Want to set it up? Just create multiple Apache server authentication records - 1 record for each host instance. In each record, a host instance is defined by a unique IP address and configuration file pair. You can create 2 records for the same IP address, but the config file can't be the same in the 2 records.

Maximum Authentication Records per API Request

A maximum of 1,000 authentication records can be processed per request, when requesting records to view, update or delete in batch mode. If the requested list identifies more than 1,000 authentication records, then the XML output includes the <WARNING> element and instructions for making another request for the next batch of records.

User Permissions

See "User Permissions and Authentication Records" for information.

Input Parameters

The input parameters for the Server authentication record API are described below.

Server Record: List

Specify a request that returns a list of application server authentication records. See "Authentication Record List by Type" for details.

Server Record: Create and Update

These parameters are used to create and update a Server authentication record.

Parameter	Description
action=create|update	(Required) Specify "create" to create a new record, or "update" to update an existing record.
echo_request={**0**|1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When unspecified, parameters are not included in the XML output.
title={value}	(Required) The title of the Server record. The title must be unique and may include a maximum of 255 characters (ascii).
comments={value}	(Optional) User defined notes about the Server record. The comments may include a maximum of 1999 characters (ascii); if comments have 2000 or more characters an error is returned and comments are not saved. Tags (such as <script>) cannot be included; if tags are included an error is returned and the request fails.
unix_apache_config_file={value}	(Required to create an Apache Web Server record; valid only for this record). The path to the Apache configuration file.
unix_apache_control_command={value}	(Required to create an Apache Web Server record; valid only for this record). The path to the Apache control command. For IBM HTTP Server, enter the path to the IBM HTTP Server "bin" directory or the specific location of "apachectl". For VMware vFabric Web Server, enter the path to the VMware vFabric global "bin" directory or the specific location of "httpdctl" for a web server instance.
unix_install_dir={value}	(Required to create an IBM WebSphere App Server record; valid only for this record). The directory where the WebSphere application is installed.
installation_path={value}	(Required to create a Tomcat Server record; valid only for this record). The directory where the tomcat server is installed. Examples: /opt/apache-tomcat-7.0.57 (e.g. $CATALINA_HOME) /opt/vmware/vfabric-tc-server-standard /opt/pivotal/pivotal-tc-server-standard

Parameter	Description
instance_path={value}	(Optional to create/update a Tomcat Server record; valid only for this record). The directory where the tomcat server instance(s) are installed. You can specify a single tomcat instance (use with **auto_discover_instances=0**), or multiple instances (use with **auto_discover_instances=1**). Leave unspecified when the instance directory is the same as the installation directory or when your targets have different types of tomcat servers. Examples: /opt/apache-tomcat-7.0.57 (e.g. $CATALINA_BASE) /opt/vmware/vfabric-tc-server-standard/tc1 /opt/pivotal/pivotal-tc-server-standard/tc1
auto_discover_instances={**0**|1}	(Optional to create/update a Tomcat Server record; valid only for this record). Specify **auto_discover_instances=1** and we'll find all tomcat server instances for you. Applies to VMware vFabric and Pivotal when you've specified a directory with multiple instances or you did not specify an instance. When unspecified (**auto_discover_instances=0**), we will not auto discover instances. Applies to Apache Tomcat or when you've specified a single instance.
ips={value}	(Required to create a record) Add IP addresses of the hosts you want to scan using this record.
ids={value}	(Required to update a record; invalid when creating a record) Update Server records with certain IDs and/or ID ranges. Multiple entries are comma separated. One or more IDs/ranges may be specified. An ID range entry is specified with a hyphen (for example, 1359-1407). Valid IDs are required.
add_ips={value}	(Optional and valid only when updating a record; invalid when creating a record). Add IP address(es) to the IP list for an existing record. You may enter a combination of IPs and IP ranges. Multiple entries are comma separated.
remove_ips={value}	(Optional and valid only when updating a record; invalid when creating a record). Remove IP address(es) from the IP list for an existing record. You may enter a combination of IPs and IP ranges. Multiple entries are comma separated.
network_id={value}	(Optional and valid when the networks feature is enabled). The network ID for the record.

Server Record: Delete

These parameters are used to delete an existing Server authentication record.

Parameter	Description
action=delete	(Required) The action required for the API request: delete.
echo_request={**0**|1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output.
ids={value}	(Required) Delete only Server records with certain IDs and/or ID ranges. Multiple entries are comma separated. One or more IDs/ranges may be specified. An ID range entry is specified with a hyphen (for example, 3000-3250). Valid IDs are required.

Server Record: XML Output

The XML output returned by a Server record API request (create, edit or delete) uses the DTD at the following URL (where qualysapi.qualys.com is the API server URL where your account is located):

```
https://qualysapi.qualys.com/api/2.0/batch_return.dtd
```

Want to learn more? Please see Appendix D.

Server Record: Sample API Request

Create an Apache record:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d
"action=create&title=Apache+Record&unix_apache_config_file=/opt/IBM/H
TTPServer/conf/httpd.conf1&unix_apache_control_command=/opt/IBM/HTTPS
erver/bin2&ips=10.10.25.25"
"https://qualysapi.qualys.com/api/2.0/fo/auth/apache/"
```

Update an Apache record:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d
"action=update&ids=1234&unix_apache_config_file=/opt/IBM/HTTPServer/c
onf/httpd.conf2"
"https://qualysapi.qualys.com/api/2.0/fo/auth/apache/"
```

HTTP Record

You have the option to choose HTTP authentication for vulnerability scans using Qualys Vulnerability Management (VM). Use the HTTP authentication API (resource **`/api/2.0/fo/auth/http/`**) for scanning protected portions of web sites and devices like printers and routers that require HTTP protocol level authentication. (Note this is not Form-based authentication). By authenticating we can perform additional vulnerability tests that we couldn't do otherwise.

How it works – During a vulnerability scan, if we come across a web page that requires HTTP authentication then we'll check to see if an HTTP record exists in your account with applicable credentials. If yes, we'll use the credentials in the record to perform HTTP authentication.

Permissions – See "User Permissions and Authentication Records".

HTTP Record: Requests

Use the parameters below to make requests to list, create, update and delete HTTP records..

Parameter	Description
action={value}	(Required) An action for the request. One of: list - list HTTP records (GET or POST) create - create a new HTTP record (POST) update - update 1 or more HTTP records (POST) delete - delete 1 or more HTTP records (POST)
title={value}	(Required for a create request; Optional for an update request; otherwise invalid) The HTTP record title.
username={value}	(Required for a create request; Optional for update request; otherwise invalid). The user name to be used for authentication.
password={value}	(Required for a create request; Optional for update request; otherwise invalid) The password to be used for authentication.
vhost={value} - or - realm={value}	(Required for create request; Optional for update request) Specify the protected device or web page you want to authenticate against. You can specify a virtual host (an FQDN such as vhost=bank.qualys.com) or the name of a realm (realm=My+Homepage).
ssl={**0** | 1}	(Optional for create or update request; otherwise invalid) Specify 1 if you want to attempt authentication over SSL only. In this case authentication is attempted only when the form is submitted via a link that uses https://...

Parameter	Description
comments={value}	(Optional for create or update request) User-defined comments.
ids={value}	(Required for update or delete request; Optional for list request; otherwise invalid) One or more HTTP record IDs.

HTTP Record: XML Output

The XML output returned by a request to configure an HTTP record (create, edit or delete) uses this DTD:

```
https://qualysapi.qualys.com/api/2.0/batch_return.dtd
```

The XML output returned by a request to list HTTP records uses this DTD:

```
https://qualysapi.qualys.com/api/2.0/fo/auth/http/auth_http_list_output.dtd
```

Want to learn more? Please see Appendix D.

HTTP Record: Sample API Requests

Create a new HTTP record - realm

API request:

```
curl -k -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d
"action=create&username=jsmith&password=abc123&title=My+HTTP+Record+1&rea
lm=My+Homepage" "https://qualysapi.qualys.com/api/2.0/fo/auth/http/"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2014-01-03T07:51:48Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>55111</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN>
```

Create a new HTTP record - virtual host

API request:

```
curl -k -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d
"action=create&username=jsmith&password=abc123&title=My+HTTP+Record+2&vho
st=bank.us.corp1.com"
"https://qualysapi.qualys.com/api/2.0/fo/auth/http/"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BATCH_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/batch_return.dtd">
<BATCH_RETURN>
  <RESPONSE>
    <DATETIME>2014-01-03T08:02:44Z</DATETIME>
    <BATCH_LIST>
      <BATCH>
        <TEXT>Successfully Created</TEXT>
        <ID_SET>
          <ID>55112</ID>
        </ID_SET>
      </BATCH>
    </BATCH_LIST>
  </RESPONSE>
</BATCH_RETURN
```

List HTTP records

API request:

```
curl -k -u "USERNAME:PASSWORD" -H "X-Requested-With: curl" -d
"action=list&ids=55111"
"https://qualysapi.qualys.com/api/2.0/fo/auth/http/"
```

XML output:

```
?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_HTTP_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/auth/http/auth_http_list_out
put.dtd">
<AUTH_HTTP_LIST_OUTPUT>
  <RESPONSE>
    <DATETIME>2014-01-03T08:08:19Z</DATETIME>
    <AUTH_HTTP_LIST>
      <AUTH_HTTP>
        <ID>55111</ID>
        <TITLE><![CDATA[My HTTP Record]]></TITLE>
```

```
          <USERNAME><![CDATA[jsmith]]></USERNAME>
          <SSL>0</SSL>
          <REALM><![CDATA[My Homepage]]></REALM>
          <CREATED>
            <DATETIME>2014-01-03T07:51:48Z</DATETIME>
            <BY>acme_ab1</BY>
          </CREATED>
          <LAST_MODIFIED>
            <DATETIME>2014-01-03T07:51:48Z</DATETIME>
          </LAST_MODIFIED>
        </AUTH_HTTP>
      </AUTH_HTTP_LIST>
    </RESPONSE>
</AUTH_HTTP_LIST_OUTPUT>
```

MySQL Record

The MySQL Authentication API (**`/api/2.0/fo/auth/mysql/`**) lets you to list, create, update and delete MySQL authentication records. User permissions for this API are the same as other authentication record APIs. Want to know more? Check out "User Permissions and Authentication Records."

List MySQL records

Use these parameters:

Parameter	Description
action=list	(Required) GET or POST method may be used.
ids={value}	(Optional) One or more IDs for MySQL records. Multiple IDS are comma separated.
id_min={value}	(Optional) Show MySQL authentication records with IDs greater than or equal to the record ID you specify.
id_max={value}	(Optional) Show MySQL records with IDs less than or equal to the record ID you specify.
title={value}	(Optional) Show records with a certain string in the record's title.
comments={value}	(Optional) Show records with a certain string in the record's comments.
details=All | **`Basic`** | None	(Optional) Show the requested amount of information about each record. A valid value is: Basic (default) - show the record ID and all record attributes None - show the record ID only All - show the record ID, all record attributes, and glossary with owner's name and login

API request:

```
curl -u USERNAME:PASSWORD -H "X-Requested-With: Curl"
"https://qualysapi.qualys.com/api/2.0/fo/auth/mysql/?action=list&details=
All"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_VAULT_LIST_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/auth/mysql/auth_mysql_list_outpu
t.dtd">
<AUTH_MYSQL_LIST_OUTPUT>
  <RESPONSE>
```

```
    <DATETIME>2015-02-02T19:31:42Z</DATETIME>
    <AUTH_MYSQL_LIST>
      <AUTH_MYSQL>
        <ID>2040156540</ID>
        <TITLE><![CDATA[MY SQL RedHat 5 CentOS 6]]></TITLE>
        <USERNAME><![CDATA[root]]></USERNAME>
        <DATABASE><![CDATA[mysql]]></DATABASE>
        <PORT>3307</PORT>
        <IP_SET>
          <IP>10.10.33.226</IP>
        </IP_SET>
        <SSL_VERIFY>false</SSL_VERIFY>
        <WINDOWS_CONF_FILE><![CDATA[C:\Program Files\MySQL\MySQL Server
5.5\test\my.ini]]></WINDOWS_CONF_FILE>
        <UNIX_CONF_FILE><![CDATA[/etc/my.cnf]]></UNIX_CONF_FILE>
        <CREATED>
          <DATETIME>2015-01-27T22:28:51Z</DATETIME>
          <BY>seenu_as</BY>
        </CREATED>
        <LAST_MODIFIED>
          <DATETIME>2015-01-29T20:13:24Z</DATETIME>
        </LAST_MODIFIED>
      </AUTH_MYSQL>
      <AUTH_MYSQL>
        <ID>2040166540</ID>
        <TITLE><![CDATA[MySQL (Unix)]]></TITLE>
        <USERNAME><![CDATA[root]]></USERNAME>
        <DATABASE><![CDATA[mysql]]></DATABASE>
        <PORT>3306</PORT>
        <IP_SET>
          <IP>10.10.10.76</IP>
          <IP>10.10.26.238</IP>
          <IP>10.10.30.132</IP>
          <IP>10.10.32.121</IP>
        </IP_SET>
        <SSL_VERIFY>false</SSL_VERIFY>
        <WINDOWS_CONF_FILE><![CDATA[]]></WINDOWS_CONF_FILE>
        <UNIX_CONF_FILE><![CDATA[/etc/my.cnf]]></UNIX_CONF_FILE>
        <CREATED>
          <DATETIME>2015-01-27T23:17:45Z</DATETIME>
          <BY>seenu_as</BY>
        </CREATED>
        <LAST_MODIFIED>
          <DATETIME>2015-01-27T23:17:45Z</DATETIME>
        </LAST_MODIFIED>
      </AUTH_MYSQL>
    </AUTH_MYSQL_LIST>
</AUTH_MYSQL_LIST_OUTPUT>
```

Create / Update MySQL records

Use these parameters:

Parameter	Description
action=create | update	(Required) POST method may be used.
title={value}	(Required for create request) The title for the new MySQL record. The title must be unique and must contain 255 characters (ascii).
username={value}	(Required for create request) The username to be used for authentication to MySQL server.
ips={value}	(Required for create request) The IP address(es) the server will log into using the record's credentials. Multiple entries are comma separated. (Optional for update request) IPs specified will overwrite existing IPs in the record, and existing IPs will be removed.
ids={value}	(Required for update request; invalid for create request) The IDs of the MySQL authentication records that you want to update. Multiple IDs are comma separated
add_ips={value}	(Optional for update request; invalid for create request) Add IPs to the IPs list for this record. Multiple IPs/ranges are comma separated.
remove_ips={value}	(Optional for update request; invalid for create request) Remove IPs from the IPs list for this record. Multiple IPs/ranges are comma separated.
password={value}	(Optional) The password to be used for authentication to MySQL server.
comments={value}	(Optional) User-defined comments. The comments may include a maximum of 1999 characters (ascii); if comments have 2000 or more characters an error is returned and comments are not saved. Tags (such as <script>) cannot be included.
ssl_verify={**0** | 1}	(Optional and valid for server that supports SSL) Specify 1 for a complete SSL certificate validation. - If unspecified (or ssl_verify=0), Qualys scanners authenticate with MySQL Servers that don't use SSL or MySQL servers that use SSL. However, in the SSL case, the server SSL certificate verification will be skipped. - If ssl_verify=1, the Qualys scanners will only send a login request after verifying that a connection the MySQL server uses SSL, the server SSL certificate is valid and matches the scanned host.

Parameter	Description
hosts={value}	(Optional) A list of FQDNs for the hosts that correspond to all host IP addresses on which a custom SSL certificate signed by a trusted root CA is installed. Multiple hosts are comma separated.
database={value}	(Optional) The database name to authenticate to. Specify a valid MySQL database name. The default is "mysql".
port={value}	(Optional) The port the database name is running on. The default is 3306.
windows_config_file={value}	(Optional) The path to the Windows MySQL config file. Access to this config file is required to run certain checks on Windows hosts.
unix_config_file={value}	(Optional) The path to the Unix MySQL config file. Access to this config file is required to run certain checks on Unix hosts.
client_cert={value}	(Optional) PEM-encoded X.509 certificate. Specify if certificate authentication is required by your server to establish an SSL connection.
client_key={value}	(Optional) PEM-encoded RSA private key. Specify if certificate authentication is required by your server to establish an SSL connection.
network_id={value}	(Optional and valid when the networks feature is enabled) The network ID for the record.

API request (create new record):

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d
"action=create&title=NewMySQLRecord&username=USERNAME&password=PASSWORD&i
ps=10.10.10.2-10.10.10.4&unix_config_file=/xyz/config/mysql.config"
"https://qualysapi.qualys.com/api/2.0/fo/auth/mysql/"
```

XML output:

```
<BATCH_RETURN>
    <RESPONSE>
        <DATETIME>2015-02-03T17:08:34Z</DATETIME>
        <BATCH_LIST>
            <BATCH>
                <TEXT>Successfully Created</TEXT>
                <ID_SET>
                    <ID>137296922</ID>
                </ID_SET>
            </BATCH>
        </BATCH_LIST>
```

```
    </RESPONSE>
</BATCH_RETURN>
```

API request (update record):

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d
"action=update&ids=137296922&password=NEWPASSWORD"
"https://qualysapi.qualys.com/api/2.0/fo/auth/mysql/"
```

XML output:

```
<BATCH_RETURN>
    <RESPONSE>
        <DATETIME>2015-01-23T17:14:28Z</DATETIME>
        <BATCH_LIST>
            <BATCH>
                <TEXT>Successfully Updated</TEXT>
                <ID_SET>
                    <ID>137296922</ID>
                </ID_SET>
            </BATCH>
        </BATCH_LIST>
    </RESPONSE>
</BATCH_RETURN>
```

Delete MySQL records

Use these parameters:

Parameter	Description
action=delete	(Required) POST method may be used.
ids={value}	(Required) MySQL authentication record IDs for the records you want to delete. Multiple records are comma separated.

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -X "POST" -d
"action=delete&ids=137296922"
"https://qualysapi.qualys.com/api/2.0/fo/auth/mysql/"
```

XML output:

```
<BATCH_RETURN>
    <RESPONSE>
        <DATETIME>2015-02-03T17:14:28Z</DATETIME>
        <BATCH_LIST>
```

```
            <BATCH>
                <TEXT>Successfully Deleted</TEXT>
                <ID_SET>
                    <ID>137296922</ID>
                </ID_SET>
            </BATCH>
        </BATCH_LIST>
    </RESPONSE>
</BATCH_RETURN>
```

MS SQL Record (compliance scans only)

The **/api/2.0/fo/auth/ms_sql/** resource allows you to manage MS SQL Server authentication records for authenticated compliance scanning (i.e. compliance scans only). You can submit API requests to view MS SQL Server authentication records, add new records, update records and delete records. Authentication is required for each API request.

See "User Permissions and Authentication Records" for information.

Before You Begin

Before you create authentication records, you must configure authentication credentials on target hosts. Please refer to the *MS SQL Server Trusted Scanning: Setup for Compliance Scans* document for further information. This document may be downloaded (in PDF) when you are logged into your Qualys account. Go to Help > Resources > Tips and Techniques and click the download link provided.

MS SQL Record: Request

URL. Use this URL to make an MS SQL Server authentication record request (create, update, or delete), where <qualysapi.qualys.com> is the API server URL where your account is located. Replace the API server URL if your account is somewhere else.

```
https://qualysapi.qualys.com/api/2.0/fo/auth/ms_sql
```

Method. For a list request, the GET or POST method may be used. For another request (create, update, or delete), the POST access method must be used.

Authentication. Authentication is required for each API request. See Chapter 2, "Authentication Using the V2 APIs."

MS SQL Record: List

Specify a request that returns a list of MS SQL Server authentication records. See "Authentication Record List by Type" for details.

MS SQL Record: Create

The parameters described below are used to specify a request that creates a new MS SQL Server authentication record.

Parameter	Description
action=create	(Required) Specifies the create action type for the request.
echo_request={**0**|1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.
title={value}	(Required) Specifies a title for the authentication record. The title must be unique and may include a maximum of 255 characters (ascii).
comments={value}	(Optional) Specifies user defined notes about the authentication record. The comments may include a maximum of 1999 characters (ascii); if comments have 2000 or more characters an error is returned and comments are not saved. Tags (such as <script>) cannot be included; if tags are included an error is returned and the request fails.
{login credentials}	Define login credentials for authentication to target hosts. See "MS SQL Record Properties: Login Credentials."
{target hosts}	Define the target hosts for authentication. See "MS SQL Record Properties: Target Hosts."
network_id={value}	(Optional and valid when the networks feature is enabled) The network ID for the record.

MS SQL Record: Update

The parameters described below are used to specify a request to update one or more existing MS SQL Server authentication records. Parameters specified in an update request will overwrite any existing parameters (previously defined).

Parameter	Description
action=update	(Required) Specifies the update action type for the request.
echo_request={**0**|1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.

Parameter	Description
ids={value}	(Required) Update only authentication records with certain IDs and/or ID ranges. Multiple entries are comma separated. One or more IDs/ranges may be specified. An ID range entry is specified with a hyphen (for example, 1359-1407). Valid IDs are required.
title={value}	(Optional) Overwrites (replaces) the existing record title with a new title. The title may include a maximum of 255 characters (ascii). When multiple IDs are specified for a batch update, the title is only replaced in the first record ID in the action since the title must be unique across records. The update will fail for the remaining IDs included in the action.
comments={value}	(Optional) Specifies user defined notes about the authentication record. The comments may include a maximum of 1999 characters (ascii); if comments have 2000 or more characters an error is returned and comments are not saved. Tags (such as <script>) cannot be included; if tags are included an error is returned and the request fails.
{login credentials}	Define login credentials for authentication to target hosts. See "MS SQL Record Properties: Login Credentials."
{target hosts}	Define the target hosts for authentication. See "MS SQL Record Properties: Target Hosts."
network_id={value}	(Optional and valid when the networks feature is enabled) The network ID for the record.

MS SQL Record: Delete

Use these parameters to delete one or more MS SQL Server authentication records.

Parameter	Description
action=delete	(Required) Specifies the delete action type for the request.
echo_request={**0** | 1}	(Optional) Show (echo) the request's input parameters (names and values) in the XML output. When not specified, parameters are not included in the XML output. Specify 1 to view parameters in the XML output.

Parameter	Description
ids={value}	(Required) Delete only authentication records with certain IDs and/or ID ranges. Multiple entries are comma separated. One or more IDs/ranges may be specified. An ID range entry is specified with a hyphen (for example, 3000-3250). Valid IDs are required.

MS SQL Record Properties: Login Credentials

The following parameters are used to define the login credentials. Parameters specified in an update request will overwrite any existing parameters (previously defined).

Parameter	Description
username={value}*	(Required for create request; Optional for update request) The user account to be used for authentication. The username may include 1-128 characters (ascii).
password={value}*	(Required for create request; Optional for update request) The password corresponding to the user account defined in the record for authentication. The password may include 1-128 characters (ascii).
db_local={0|1}*	(Optional) A flag indicating the authentication type. Set to 1 when login credentials are for a MS SQL Server database account. Set to 0 when login credentials are for a Microsoft Windows operating system account that is associated with a MS SQL Server database account. For a create request if the **db_local** parameter is unspecified, the flag is set to 1.
windows_domain={value}	(Required when **db_local=0**; otherwise invalid) The domain name where the login credentials are stored when the login credentials are for a Microsoft Windows operating system account that is associated with a MS SQL Server database account. The domain name may include 1-256 characters (ascii). For an update request when the credentials for the record are for a Microsoft Windows account (**db_local=0**) and you want to change the record to use credentials for a MS SQL Server account (**db_local=1**) note the following. You must set **windows_domain=''** (the empty string) to clear the current parameter setting.

Parameter	Description
instance={value}*	(Optional) The name of the database instance to be scanned. This is the instance name assigned to the TCP/IP port. Important: This is not the host name that is assigned to the MS SQL Server instance name (see "MS SQL Server Instance Name" in the Qualys online help for information). The instance name may include a maximum of 128 characters (ascii). For a create request if the instance parameter is unspecified, the instance name is set to "MSSQLSERVER".
	These parameters are mutually exclusive: instance and auto_discover_instances=1.
auto_discover_instances={**0**|1}	Set **auto_discover_instances=1** and we'll find all MS SQL Server instance names on each host. Note Windows authentication is required in order for us to auto discover instance names - be sure you set up Windows authentication records for your hosts running MS SQL Server
	These parameters are mutually exclusive: instance and auto_discover_instances=1.
database={value}*	(Optional) The database name of the database to be scanned. The database name may contain a maximum of 128 characters. For a create request if the database name is unspecified, the database name is set to "master".
	These parameters are mutually exclusive: database and auto_discover_databases=1.
auto_discover_databases={**0**|1}	Set **auto_discover_databases=1** and we'll find all MS SQL Server database names on each host.
	These parameters are mutually exclusive: database and auto_discover_databases=1.
port={value}*	(Required for create request; Optional for update request) The port number assigned to the database instance to be scanned.
	To create a record you must specify one of these parameters: port or auto_discover_ports=1. These parameters are mutually exclusive.
auto_discover_ports={**0**|1}	Set **auto_discover_ports=1** and for each host we'll find all ports MS SQL Server is running on. Note Windows authentication is required for us to auto discover ports - be sure you set up Windows authentication records for your hosts running MS SQL Server.
	To create a record you must specify one of these parameters: port or auto_discover_ports=1. These parameters are mutually exclusive.

MS SQL Record Properties: Target Hosts

For an MS SQL authentication record, the combination of each login type (Windows or MS SQL Server), account login name, database information (instance name, database name, port) and IP addresses must be unique.

The following parameters are used to define the target compliance hosts that the scanning engine should log into with the specified credentials. Only compliance hosts can be added to an MS SQL Server authentication record.

Parameter	Description
ips={value}	(Required for create request) Defines an IP list for the authentication record. (Optional for update request) Overwrites (replaces) the IP list for the authentication record. The IPs you specify are added and any existing IPs are removed. You may enter a combination of IPs and IP ranges to identify compliance hosts. Multiple entries are comma separated. This parameter and the **add_ips** parameter or the **remove_ips** parameter cannot be specified in the same request.
add_ips={value}	(Optional for update request only) This parameter is used to update an existing IP list in an existing authentication record. Specifies one or more IP addresses to add to the IP list for the authentication record. You may enter a combination of IPs and IP ranges to identify compliance hosts. Multiple entries are comma separated. This parameter and the **ips** parameter cannot be specified in the same request.
remove_ips={value}	(Optional for update request only) This parameter is used to update an existing IP list in an existing authentication record. Specifies one or more IP addresses to remove from the IP list for an existing authentication record. You may enter a combination of IPs and IP ranges to identify compliance hosts. Multiple entries are comma separated. This parameter and the **ips** parameter cannot be specified in the same request.

MS SQL Record: XML Output

The XML output returned by an MS SQL Server record API request (create, edit or delete) uses the DTD at the following URL (where qualysapi.qualys.com is the API server URL where your account is located):

```
http://qualysapi.qualys.com/api/2.0/batch_return.dtd
```

Want to learn more? Please see Appendix D.

Authentication Vault

The Authentication Vault API (resource **`/api/2.0/fo/vault`**) allows you to manage authentication vaults for authentication records that use them. Using this API you can list vaults, create new vaults, update and view vault settings, and delete vaults.

Permissions: Managers, Unit Managers and Scanners can view vaults and their settings. Managers can perform more functions (create, update, delete). Unit Managers can perform these functions if they are granted the permission "Create/edit authentication records/vaults".

List vaults

Use the parameter "action=list" to list the vaults defined in your account.

API request:

```
curl -u "USERNAME:PASSWD" -H "X-Requested-With: curl" -d "action=list"
"https://qualysapi.qualys.com/api/2.0/fo/vault/"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE AUTH_VAULT_LIST_OUTPUT SYSTEM

"https://qualysapi.qualys.com/api/2.0/fo/vault/vault_output.dtd">
<AUTH_VAULT_LIST_OUTPUT>
    <RESPONSE>
        <DATETIME>2014-09-12T13:55:57Z</DATETIME>
        <STATUS>Success</STATUS>
        <COUNT>13</COUNT>
        <AUTH_VAULTS>
            <AUTH_VAULT>
                <TITLE>
                    <![CDATA[added failover ip]]>
                </TITLE>
                <VAULT_TYPE>
                    <![CDATA[Cyber-Ark PIM Suite]]>
                </VAULT_TYPE>
                <LAST_MODIFIED>
                    <DATETIME>2014-02-13T12:05:21Z</DATETIME>
                    <BY>quays_rn1</BY>
                </LAST_MODIFIED>
                <ID>1421</ID>
            </AUTH_VAULT>
            <AUTH_VAULT>
                <TITLE>
```

```
                    <![CDATA[added failover ip1]]>
                </TITLE>
                <VAULT_TYPE>
                    <![CDATA[Cyber-Ark PIM Suite]]>
                </VAULT_TYPE>
                <LAST_MODIFIED>
                    <DATETIME>2014-02-19T06:43:44Z</DATETIME>
                    <BY>quays_rn1</BY>
                </LAST_MODIFIED>
                <ID>1441</ID>
            </AUTH_VAULT>
            <AUTH_VAULT>
                <TITLE>
                    <![CDATA[Blue]]>
                </TITLE>
                <VAULT_TYPE>
                    <![CDATA[CA Access Control]]>
                </VAULT_TYPE>
                <LAST_MODIFIED>
                    <DATETIME>2013-09-21T05:26:32Z</DATETIME>
                    <BY>quays_rn1</BY>
                </LAST_MODIFIED>
                <ID>1406</ID>
            </AUTH_VAULT>
        </AUTH_VAULTS>
    </RESPONSE>
</AUTH_VAULT_LIST_OUTPUT>
```

Parameters:

Parameter	Description
action=list	(Required)
echo_request={**0** | 1}	(Optional) Set to 1 to show (echo) the request's input parameters (names and value) in the XML output.
title={value}	(Optional) Include vaults matching this title.
type={value}	(Optional) Include a certain vault type only. A valid value is: Cyber-Ark PIM Suite Thycotic Secret Server Quest Vault CA Access Control Hitachi ID PAM Lieberman ERPM
modified={date}	(Optional) Include vaults modified on or after a certain date/time, in this format: YYYY-MM-DD[THH:MM:SSZ] (UTC/GMT).

Parameter	Description
orderby={value}	(Optional) Sort the vaults list by certain data. One of: "id", "title", "system_name", "last_modified", "last_modified_by". A date must be specified in YYYYMM-DD[THH:MM:SSZ] format (UTC/GMT).
sortorder={asc | desc}	(Optional) The sort order, used when the request includes the **orderby** parameter. One of: asc (for ascending order) or desc (for descending order).
limit={value}	(Optional) The maximum number of vault records processed for the request, starting at the record number specified by the **offset** parameter. These parameters must be specified together: **limit** and **offset**. When not specified, default limit is set to 1,000 vault records. You can specify a value less than or greater than the default. It's possible to specify "limit=0" for no limit. In this case the output is not paginated and all records are returned in a single output. Warning: This is not recommended since it may generate a very large output and processing large XML files can consume a lot of resources on the client side.
offset={value}	(Optional) The starting vault record number, used only when the request includes the **limit** parameter.

More sample requests:

1) List all vaults, order vaults by system name

```
curl -H "X-Requested-With:API" -u "USERNAME:PASSWD" -d
"action=list&orderby=system_name"
"https://qualysapi.qualys.com/api/2.0/fo/vault/index.php/?"
```

2) List all vaults, order vaults by title in descending order

```
curl -H "X-Requested-With:API" -u "USERNAME:PASSWD" -d
"action=list&sortorder=desc&title"
"https://qualysapi.eng.qualys.com/api/2.0/fo/vault/index.php/?"
```

3) List only 9th and 10th vault records

```
curl -H "X-Requested-With:API" -u "USERNAME:PASSWD" -d
"action=list&limit=2&offset=9"
"https://qualysapi.qualys.com/api/2.0/fo/vault/index.php/?"
```

Create a new vault

Use the parameter “action=create”.

Parameters:

Parameter	Description
action=create	(Required)
title={value}	(Required) The vault title.
type={value}	(Required) The vault type. A valid value is: Cyber-Ark PIM Suite Thycotic Secret Server Quest Vault CA Access Control Hitachi ID PAM Lieberman ERPM
comments={value}	(Optional) User defined comments.
{vault settings}	“Tell me about vault settings”

API request:

```
curl -u "USERNAME:PASSWD" -H "X-Requested-With: curl" -X "POST" -d
"title=My+Vault&type=Cyber-Ark+PIM+Suite&server_address=123.123.123.1
23&port=1858&safe=mysafe1&username=someusername&password=somepasswd"
"https://qualysapi.qualys.com/api/2.0/fo/vault/?action=create"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
    <RESPONSE>
        <DATETIME>2014-09-12T14:13:28Z</DATETIME>
        <TEXT>Success</TEXT>
        <ITEM_LIST>
            <ITEM>
                <KEY>ID</KEY>
                <VALUE>14836922</VALUE>
            </ITEM>
        </ITEM_LIST>
    </RESPONSE>
</SIMPLE_RETURN>
```

Update vault settings

Use the parameter “action=update”.

Parameters:

Parameter	Description
action=update	(Required)
id={value}	(Required) A vault ID.
title={value}	(Optional) A new title to replace the existing title.
comments={value}	(Optional) User defined comments.
{vault settings}	“Tell me about vault settings”

API request:

```
curl -u "USERNAME:PASSWD" -H "X-Requested-With: curl" -X "POST" -d
"id=14836922&server_address=10.10.10.10"
"https://qualysapi.qualys.com/api/2.0/fo/vault/?action=update"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
    <RESPONSE>
        <DATETIME>2014-09-12T14:13:28Z</DATETIME>
        <TEXT>Success</TEXT>
        <ITEM_LIST>
            <ITEM>
                <KEY>ID</KEY>
                <VALUE>14836922</VALUE>
            </ITEM>
        </ITEM_LIST>
    </RESPONSE>
</SIMPLE_RETURN>
```

View vault settings

Use the parameter "action=view".

Parameter	Description
action=view	(Required)
id={value}	(Required) A vault ID.

API request:

```
curl -u "USERNAME:PASSWD" -H "X-Requested-With: curl"
"id=14836922"
"https://qualysapi.qualys.com/api/2.0/fo/vault/?action=view"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE VAULT_OUTPUT SYSTEM
"https://qualysapi.qualys.com/api/2.0/fo/vault/vault_view.dtd">
<VAULT_OUTPUT>
    <RESPONSE>
        <DATETIME>2014-09-12T14:25:04Z</DATETIME>
        <VAULT_QUEST>
            <TITLE>
                <![CDATA[My vault]]>
            </TITLE>
            <COMMENTS>
                <![CDATA[Some comments]]>
            </COMMENTS>
            <VAULT_TYPE>
                <![CDATA[Thycotic Secret Server]]>
            </VAULT_TYPE>
            <CREATED_ON>2014-09-12T14:13:28Z</CREATED_ON>
            <OWNER>acme_ab1</OWNER>
            <LAST_MODIFIED>
                <DATETIME>2014-09-12T14:13:28Z</DATETIME>
                <BY>acme_ab1</BY>
            </LAST_MODIFIED>
            <USERNAME>
                <![CDATA[acme_ab1]]>
            </USERNAME>
            <URL>
                <![CDATA[https://corp1.corp.com]]>
            </URL>
            <DOMAIN>
                <![CDATA[corp-test.com]]>
```

```
            </DOMAIN>
            <ID>14836922</ID>
        </VAULT_QUEST>
    </RESPONSE>
</VAULT_OUTPUT>
```

Delete a vault

Use the parameter "action=delete".

Parameter	Description
action=view	(Required)
id={value}	(Required) A vault ID.

API request:

```
curl -u "USERNAME:PASSWD" -H "X-Requested-With: curl" -d "id=43463"
"https://qualysapi.qualys.com/api/2.0/fo/vault/?action=delete"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
    <RESPONSE>
        <DATETIME>2014-09-12T14:13:28Z</DATETIME>
        <TEXT>Success</TEXT>
        <ITEM_LIST>
            <ITEM>
                <KEY>Status</KEY>
                <VALUE>Deleted</VALUE>
            </ITEM>
        </ITEM_LIST>
    </RESPONSE>
</SIMPLE_RETURN>
```

Tell me about vault settings

The vault settings differ per vault type.

CA Access Control	
ca_url={value}	(Required for new vault) The HTTP or HTTPS URL of the CA Access Control web services, an API interface to your CA Access Control Enterprise Management installation. Note that the web services URL is different from the web management URL. Sample web services URL: http://caac126u-32-235.caac125.domain.com:18080/iam/TEWS6/ac Sample web management URL: http://caac126p-33-166.caac125.domain.com:18080/iam/ac/
ca_api_username={value}	(Required for new vault) The name of a user that is granted GetAccountPassword API permissions.
ca_ssl_verify={1|0}	(Required for new vault) When set to 1, our service will verify the SSL certificate of the web server to make sure the certificate is valid and trusted. When set to 0 our service will not verify the certificate of the web server.
ca_web_username={value}	(Optional) The web user name used to access Basic Authentication of the CA Access Control web server.
ca_web_password={value}	(Optional) The web password used to access Basic Authentication of the CA Access Control web server.
Cyber-Ark PIM Suite	
server_address={value}	(Required for new vault) The IP address of the vault server that stores system login credentials to be used.
port={value}	(Optional) The port the vault server is running on. The port must be in the range 1025 to 65535. For a new vault the port is set to 1858 by default, if the port parameter is not specified.
safe={value}	(Required for new vault) The name of the digital password safe. The safe name can contain a maximum of 28 characters (leading and/or trailing space in the input value will be removed). These special characters cannot be included in a safe name: \ / : * ? " < > . |
username={value}	(Required for new vault) The username for an account with access to your Cyber-Ark PIM Suite environment.
password={value}	(Required for new vault) The password for an account with access to your Cyber-Ark PIM Suite environment.

Hitachi ID PAM	
url={value}	(Required for new vault) The HTTP or HTTPS URL of the Hitachi ID PAM webservices.
username={value}	(Required for new vault) The username (ID) for the Hitachi ID PAM user account. To allow Qualys scanners to connect using this account, this user must have the following settings under Administrator information in the Hitachi ID Management Suite: 1) the privilege “OTP IDAPI caller” and 2) the value entered in the “IP address with CIDR bitmask” field must include the Qualys scanner IP addresses.
password={value}	(Required for new vault) The password for the Hitachi ID PAM user account.
ssl_verify={1|0}	(Required for new vault) When set to 1, our service will verify the SSL certificate of the web server to make sure the certificate is valid and trusted. When set to 0 our service will not verify the certificate of the web server.
Lieberman ERPM	
url={value}	(Required for new vault) The HTTP or HTTPS URL of the Lieberman ERPM server.
domain={value}	(Optional) A domain name if your Lieberman ERPM server is part of a domain.
username={value}	(Required for new vault) The username for the Lieberman ERPM server account.
password={value}	(Required) The password for the Lieberman ERPM server account.
ssl_verify={1|0}	(Required for new vault) When set to 1, our service will verify the SSL certificate of the web server to make sure the certificate is valid and trusted. When set to 0 our service will not verify the certificate of the web server.
Quest Vault	
server_address={value}	(Required for new vault) The IP address of the vault server, Quest One Privileged Password Manager.
port={value}	(Optional) The listing port of the vault server. For a new vault the port is set to 22 by default, if the port parameter is not specified.
username={value}	(Required for new vault) The username to be used for SSH authentication. We recommend you create a dedicated user account for Qualys scanning. Using Quest/Dell 2.4 or higher, enter the key for the API user account you've created for use with our service. We support both API and CLI keys but recommend use of an API key.

access_key={value}	(Required for new vault) The DSA private key in PEM format for SSH authentication.
Thycotic Secret Server	
url={value}	(Required for new vault) The HTTP or HTTPS URL of the Secret Server webservices. The URL may contain a maximum of 256 characters, and must not contain multibyte characters.
username={value}	(Required for new vault) The username for a Secret Server user. This user must have access to the secret names to be used for authentication.
password={value}	(Required for new vault) The password for a Secret Server user.
domain={value}	(Optional) Specify a fully qualified domain name if Secret Server is integrated with Active Directory. The domain may contain a maximum of 128 characters, and must not contain any multibyte characters.

CHAPTER 9

Network API

The Network API is used to manage networks when the Network Support feature is enabled in the user's subscription.

These topics are covered:

- Network List
- Create Network
- Update Network
- Assign Scanner Appliance to Network

Network List

The Network List API v2 (resource /api/2.0/fo/network/ with parameter action=list) is used to list custom networks in your account. The optional "ids" input parameter can be used to filter the list.

Supported methods	GET and POST
Permissions	This API is available to all users with the API access permission. A Manager will view all custom networks in the subscription, a Unit Manager will view custom networks in their business unit's assigned asset groups, and a Scanner/Reader will view custom networks in their account's assigned asset groups.

Sample

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl"
"https://qualysapi.qualys.com/api/2.0/fo/network/?action=list&ids=734
3,7345,7350"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE NETWORK_LIST SYSTEM
"https://qualysapi.qualys.com/network_list_output.dtd">
<RESPONSE>
  <DATETIME>2013-07-28T01:06:45Z</DATETIME>
  <NETWORK_LIST>
    <NETWORK>
      <ID>7343</ID>
      <NAME><![CDATA[My New Network]]></TITLE>
      <SCANNER_APPLIANCE_LIST>
        <SCANNER_APPLIANCE>
          <ID>1234</ID>
          <FRIENDLY_NAME><![[CDATA[abc123]]></FRIENDLY_NAME>
        </SCANNER_APPLIANCE>
      </SCANNER_APPLIANCE_LIST>
    </NETWORK>
    ...
  </NETWORK_LIST>
</RESPONSE>
```

Create Network

The Create Networlk API v2 (resource /api/2.0/fo/network/ with action=create) is used to create a new network. The input parameter "name" (required) is a user-defined friendly name. A successful request will return a unique network ID and this is used to manage your network using the API.

Supported methods	POST
Permissions	This API is available to Managers only

Next steps

Before you're ready to start scanning, you'll need to 1) assign scanner appliance(s) to your network, and 2) add host assets to your network (assign asset groups to it).

Sample

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -d
"action=create&name=My+Network"
"https://qualysapi.qualys.com/api/2.0/fo/network/"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2014-01-14T04:37:24Z</DATETIME>
    <TEXT>Network created with ID</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>id</KEY>
        <VALUE>1103</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

Update Network

The Update Network API v2 (resource /api/2.0/fo/network/ with parameter action=update) is used to change the name for a network. Use the "name" parameter to specify a new network name. (The network ID is assigned by our service and it can't be changed.)

Supported methods	POST
Permissions	This API is available to Managers only

Sample

API request:

```
curl -u "USERNAME:PASSWORD" -H "X-Requested-With: Curl" -d
"id=1130&action=update&name=Network+123"
"https://qualysapi.qualys.com/api/2.0/fo/network/"
```

XML output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2014-01-20T06:17:06Z</DATETIME>
    <TEXT>Network updated</TEXT>
    <ITEM_LIST>
      <ITEM>
        <KEY>id</KEY>
        <VALUE>1103</VALUE>
      </ITEM>
      <ITEM>
        <KEY>name</KEY>
        <VALUE>Network 123</VALUE>
      </ITEM>
    </ITEM_LIST>
  </RESPONSE>
</SIMPLE_RETURN>
```

Assign Scanner Appliance to Network

The Assign Scanner Appliance to Network API v2 (resource /api/2.0/fo/appliance/ with action=assign_network_id) is used to assign a scanner appliance to a network. When the network support feature is enabled for your subscription, scanner appliances are assigned to networks. Each appliance can be assigned to 1 network only.

Supported methods	POST
Permissions	This API is available to Managers only
Required input parameters	action=assign_network_id appliance_id={id} network_id={id}

Sample

API request:

```
curl -k -u "USERNAME:PASSWORD" -H "X-Requested-With: test" -d
action=assign_network_id&appliance_id=506&network_id=1002"
"https://qualysapi.qualys.com/api/2.0/fo/appliance/"
```

XML output:

The simple return DTD is used. The response will look like this, if successful:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2013-12-16T22:50:49Z</DATETIME>
    <TEXT>Success: Network ID=[1103] assigned to Appliance with
ID=[506]</TEXT>
  </RESPONSE>
</SIMPLE_RETURN>
```

Or, if unsuccessful, the response might look like this:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SIMPLE_RETURN SYSTEM
"https://qualysapi.qualys.com/api/2.0/simple_return.dtd">
<SIMPLE_RETURN>
  <RESPONSE>
    <DATETIME>2013-12-16T22:53:41Z</DATETIME>
    <CODE>1905</CODE>
    <TEXT>parameter network_id has invalid value: 1103 (No such
```

```
network ID)</TEXT>
  </RESPONSE>
</SIMPLE_RETURN>
```

APPENDIX A

Scan and Report XML

This appendix describes the XML output returned from API V2 requests.

The following topics are included:

- Simple Return
- Batch Return
- Scan List Output
- SCAP Scan List Output
- Scheduled Scan List Output
- Vulnerability Scan Results
- Compliance Scan Results
- PCI Scan Share Status Output
- Scanner Appliance List Output
- Scanner Appliance Create Output
- KnowledgeBase Output
- Report List
- Network List

Simple Return

The simple return is XML output returned from several API V2 calls. The DTD "simple_return.dtd" can be found at the following URL:

```
https://qualysapi.qualys.com/api/2.0/simple_return.dtd
```

where <qualysapi.qualys.com> is the API server URL where your account is located.

DTD for Simple Return

A recent DTD for the simple return output (simple_return.dtd) is shown below.

```
<!-- QUALYS SIMPLE_RETURN DTD -->

<!ELEMENT SIMPLE_RETURN (REQUEST?, RESPONSE)>

<!ELEMENT REQUEST (DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?,
                   POST_DATA?)>
<!ELEMENT DATETIME (#PCDATA)>
<!ELEMENT USER_LOGIN (#PCDATA)>
<!ELEMENT RESOURCE (#PCDATA)>
<!ELEMENT PARAM_LIST (PARAM+)>
<!ELEMENT PARAM (KEY, VALUE)>
<!ELEMENT KEY (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>
<!-- If specified, POST_DATA will be urlencoded -->
<!ELEMENT POST_DATA (#PCDATA)>

<!ELEMENT RESPONSE (DATETIME, CODE?, TEXT, ITEM_LIST?)>
<!ELEMENT CODE (#PCDATA)>
<!ELEMENT TEXT (#PCDATA)>
<!ELEMENT ITEM_LIST (ITEM+)>
<!ELEMENT ITEM (KEY, VALUE*)>
```

XPaths for Simple Return

This section describes the XPaths for the simple return output (simple_return.dtd).

XPath	element specifications / notes
/SIMPLE_RETURN	(REQUEST?, RESPONSE)
/SIMPLE_RETURN/REQUEST	(DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?, POST_DATA?)
/SIMPLE_RETURN/REQUEST/DATETIME	(#PCDATA)
	The date and time of the request.
/SIMPLE_RETURN/REQUEST/USER_LOGIN	(#PCDATA)
	The user login ID of the user who made the request.
/SIMPLE_RETURN/REQUEST/RESOURCE	(#PCDATA)
	The resource specified for the request.
/SIMPLE_RETURN/REQUEST/PARAM_LIST	(PARAM+)
/SIMPLE_RETURN/REQUEST/PARAM_LIST/PARAM	(KEY, VALUE)
/SIMPLE_RETURN/REQUEST/PARAM_LIST/PARAM/KEY	(#PCDATA)
	The input parameter name.
/SIMPLE_RETURN/REQUEST/PARAM_LIST/PARAM/VALUE	(#PCDATA)
	The input parameter value.
/SIMPLE_RETURN/REQUEST/POST_DATA	(#PCDATA)
	The POST data.
/SIMPLE_RETURN/RESPONSE	(DATETIME, CODE?, TEXT, ITEM_LIST?)
/SIMPLE_RETURN/RESPONSE/DATETIME	(#PCDATA)
	The date and time of the response.
/SIMPLE_RETURN/RESPONSE/CODE	(#PCDATA)
	The response error code.
/SIMPLE_RETURN/RESPONSE/TEXT	(#PCDATA)
	The response error text.
/SIMPLE_RETURN/RESPONSE/ITEM_LIST	(ITEM+)
/SIMPLE_RETURN/RESPONSE/ITEM_LIST/ITEM	(KEY, VALUE+)
/SIMPLE_RETURN/RESPONSE/ITEM_LIST/ITEM/KEY	(#PCDATA)
	The response item keyword.
/SIMPLE_RETURN/RESPONSE/ITEM_LIST/ITEM/KEY	(#PCDATA)
	The response item value.

Batch Return

The batch return is XML output returned from several API V2 calls. The DTD "batch_return.dtd" can be found at the following URL:

```
https://qualysapi.qualys.com/api/2.0/batch_return.dtd
```

where <qualysapi.qualys.com> is the API server URL where your account is located.

DTD for Batch Return

A recent DTD for the simple return output (batch_return.dtd) is shown below.

```
<!-- QUALYS BATCH_RETURN DTD -->
<!ELEMENT BATCH_RETURN (REQUEST?,RESPONSE)>

<!ELEMENT REQUEST (DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?,
                   POST_DATA?)>
<!ELEMENT DATETIME (#PCDATA)>
<!ELEMENT USER_LOGIN (#PCDATA)>
<!ELEMENT RESOURCE (#PCDATA)>
<!ELEMENT PARAM_LIST (PARAM+)>
<!ELEMENT PARAM (KEY, VALUE)>
<!ELEMENT KEY (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>
<!-- If specified, POST_DATA will be urlencoded -->
<!ELEMENT POST_DATA (#PCDATA)>

<!ELEMENT RESPONSE (DATETIME, BATCH_LIST?)>
<!ELEMENT BATCH_LIST (BATCH+)>
<!ELEMENT BATCH (CODE?, TEXT?, ID_SET?)>

<!ELEMENT CODE (#PCDATA)>
<!ELEMENT TEXT (#PCDATA)>
<!ELEMENT ID_SET (ID|ID_RANGE)+>
<!ELEMENT ID_RANGE (#PCDATA)>
<!ELEMENT ID (#PCDATA)>
<!-- EOF -->
```

XPaths for Batch Return

This section describes the XPaths for the simple return output (batch_return.dtd).

XPath	element specifications / notes
/BATCH_RETURN	(REQUEST?, RESPONSE)
/BATCH_RETURN/REQUEST	(DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?, POST_DATA?)
/BATCH_RETURN/REQUEST/DATETIME	(#PCDATA)
	The date and time of the request.
/BATCH_RETURN/REQUEST/USER_LOGIN	(#PCDATA)
	The user login ID of the user who made the request.
/SIMPLE_RETURN/REQUEST/RESOURCE	(#PCDATA)
	The resource specified for the request.
/BATCH_RETURN/REQUEST/PARAM_LIST	(PARAM+)
/BATCH_RETURN/REQUEST/PARAM_LIST/PARAM	(KEY, VALUE)
/BATCH_RETURN/REQUEST/PARAM_LIST/PARAM/KEY	(#PCDATA)
	The input parameter name.
/BATCH_RETURN/REQUEST/PARAM_LIST/PARAM/VALUE	(#PCDATA)
	The input parameter value.
/BATCH_RETURN/REQUEST/POST_DATA	(#PCDATA)
	The POST data.
/BATCH_RETURN/RESPONSE	(DATETIME, BATCH_LIST?)
/BATCH_RETURN/RESPONSE/DATETIME	(#PCDATA)
	The date and time of the response.
/BATCH_RETURN/RESPONSE/BATCH_LIST	(BATCH+)
/BATCH_RETURN/RESPONSE/BATCH_LIST/BATCH	(CODE?, TEXT?, ID_SET?)
/BATCH_RETURN/RESPONSE/BATCH_LIST/BATCH/CODE	(#PCDATA)
	A batch code.
/BATCH_RETURN/RESPONSE/BATCH_LIST/BATCH/TEXT	(#PCDATA)
	A batch text description.
/BATCH_RETURN/RESPONSE/BATCH_LIST/BATCH/ID_SET	(ID|ID_RANGE)
/BATCH_RETURN/RESPONSE/BATCH_LIST/BATCH/ID_SET/ID	(#PCDATA)
	A batch ID number.
/BATCH_RETURN/RESPONSE/BATCH_LIST/BATCH/ID_SET/ID_RANGE	(#PCDATA)
	A batch ID range.

Scan List Output

The scan list output is an XML report returned from the VM and PC scan list API calls. The DTD "scan_list_output.dtd" can be found at the following URL (when your account is located on US Platform 1):

```
https://qualysapi.qualys.com/api/2.0/fo/scan/
scan_list_output.dtd
```

DTD for Scan List Output

A recent DTD for the scan list output (scan_list_output.dtd) is shown below.

```
<!-- QUALYS SCAN_LIST_OUTPUT DTD -->
<!ELEMENT SCAN_LIST_OUTPUT (REQUEST?,RESPONSE)>
<!ELEMENT REQUEST (DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?,
                   POST_DATA?)>
<!ELEMENT DATETIME (#PCDATA)>
<!ELEMENT USER_LOGIN (#PCDATA)>
<!ELEMENT RESOURCE (#PCDATA)>
<!ELEMENT PARAM_LIST (PARAM+)>
<!ELEMENT PARAM (KEY, VALUE)>
<!ELEMENT KEY (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>
<!-- if returned, POST_DATA will be urlencoded -->
<!ELEMENT POST_DATA (#PCDATA)>

<!ELEMENT RESPONSE (DATETIME, SCAN_LIST?)>
<!ELEMENT SCAN_LIST (SCAN+)>
<!ELEMENT SCAN (ID?, REF, TYPE, TITLE, USER_LOGIN, LAUNCH_DATETIME,
               DURATION, PROCESSED, STATUS?, TARGET,
               ASSET_GROUP_TITLE_LIST?, OPTION_PROFILE?)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT REF (#PCDATA)>
<!ELEMENT TYPE (#PCDATA)>
<!ELEMENT TITLE (#PCDATA)>
<!ELEMENT LAUNCH_DATETIME (#PCDATA)>
<!ELEMENT DURATION (#PCDATA)>
<!ELEMENT PROCESSED (#PCDATA)>
<!ELEMENT STATUS (STATE, SUB_STATE?)>
<!ELEMENT STATE (#PCDATA)>
<!ELEMENT SUB_STATE (#PCDATA)>
<!ELEMENT TARGET (#PCDATA)>
<!ELEMENT ASSET_GROUP_TITLE_LIST (ASSET_GROUP_TITLE+)>
<!ELEMENT ASSET_GROUP_TITLE (#PCDATA)>
<!ELEMENT OPTION_PROFILE (TITLE, DEFAULT_FLAG?)>
<!ELEMENT DEFAULT_FLAG (#PCDATA)>
<!-- EOF -->
```

XPaths for Scan List Output

This section describes the XPaths for the scan list output (scan_list_output.dtd).

XPath	element specifications / notes
/SCAN_LIST_OUTPUT	(REQUEST?, RESPONSE)
/SCAN_LIST_OUTPUT/REQUEST	(DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?, POST_DATA?)
/SCAN_LIST_OUTPUT/REQUEST/DATETIME	(#PCDATA) The date and time of the request.
/SCAN_LIST_OUTPUT/REQUEST/USER_LOGIN	(#PCDATA) The user login ID of the user who made the request.
/SCAN_LIST_OUTPUT/REQUEST/RESOURCE	(#PCDATA) The resource specified for the request.
/SCAN_LIST_OUTPUT/REQUEST/PARAM_LIST	(PARAM+)
/SCAN_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM	(KEY, VALUE)
/SCAN_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM/KEY	(#PCDATA) The input parameter name.
/SCAN_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM/VALUE	(#PCDATA) The input parameter value.
/SCAN_LIST_OUTPUT/REQUEST/POST_DATA	(#PCDATA) The POST data, if any.
/SCAN_LIST_OUTPUT/RESPONSE	(DATETIME, SCAN_LIST?)
/SCAN_LIST_OUTPUT/RESPONSE/SCAN_LIST	(SCAN+)
/SCAN_LIST_OUTPUT/RESPONSE/SCAN_LIST/SCAN	(ID?, REF, TYPE, TITLE, USER_LOGIN, LAUNCH_DATETIME, DURATION, PROCESSED, STATUS?, TARGET, ASSET_GROUP_TITLE_LIST?, OPTION_PROFILE?)
/SCAN_LIST_OUTPUT/RESPONSE/SCAN_LIST/SCAN/ID	(#PCDATA) The scan ID.
/SCAN_LIST_OUTPUT/RESPONSE/SCAN_LIST/SCAN/REF	(#PCDATA) The scan reference code.
/SCAN_LIST_OUTPUT/RESPONSE/SCAN_LIST/SCAN/TYPE	(#PCDATA) The scan type: On-Demand, Scheduled or API.
/SCAN_LIST_OUTPUT/RESPONSE/SCAN_LIST/SCAN/TITLE	(#PCDATA) The scan title.
/SCAN_LIST_OUTPUT/RESPONSE/SCAN_LIST/SCAN/USER_LOGIN	(#PCDATA) The user login ID of the user who launched the scan.

XPath	element specifications / notes
/SCAN_LIST_OUTPUT/RESPONSE/SCAN_LIST/SCAN/LAUNCH_DATETIME (#PCDATA)	
	The date and time when the scan was launched.
/SCAN_LIST_OUTPUT/RESPONSE/SCAN_LIST/SCAN/DURATION (#PCDATA)	
	The time it took to perform the scan - when the scan status is Finished. For a scan that has not finished (queued, running), the duration is set to “Pending”.
/SCAN_LIST_OUTPUT/RESPONSE/SCAN_LIST/SCAN/PROCESSED (#PCDATA)	
	A flag that specifies whether the scan results have been processed. A value of 1 is returned when the scan results have been processed. A value of 0 is returned when the results have not been processed.
/SCAN_LIST_OUTPUT/RESPONSE/SCAN_LIST/SCAN/STATUS	
	(STATE, SUB-STATE?)
/SCAN_LIST_OUTPUT/RESPONSE/SCAN_LIST/SCAN/STATUS/STATE (#PCDATA)	
	The scan state: Running, Paused, Canceled, Finished, Error, Queued (scan job is waiting to be distributed to scanner(s)), or Loading (scanner(s) are finished and scan results are being loaded onto the platform).
/SCAN_LIST_OUTPUT/RESPONSE/SCAN_LIST/SCAN/STATUS/SUB_STATE (#PCDATA)	
	The sub-state related to the scan state, if any. For scan state Finished, value can be: No_Vuln (no vulnerabilities found) or No_Host (no host alive). For scan state Queued, value can be: Launching (service received scan request), Pausing (service received pause scan request), or Resuming (service received resume scan request).
/SCAN_LIST_OUTPUT/RESPONSE/SCAN_LIST/SCAN/TARGET (#PCDATA)	
	The scan target hosts.
/SCAN_LIST_OUTPUT/RESPONSE/SCAN_LIST/SCAN/ASSET_GROUP_TITLE_LIST (ASSET_GROUP_TITLE+)	
/SCAN_LIST_OUTPUT/RESPONSE/SCAN_LIST/SCAN/ASSET_GROUP_TITLE_LIST/ASSET_GROUP_TITLE	
	(#PCDATA)
	The asset group title specified for the scan.
/SCAN_LIST_OUTPUT/RESPONSE/SCAN_LIST/SCAN/OPTION_PROFILE (TITLE, DEFAULT_FLAG?)	
/SCAN_LIST_OUTPUT/RESPONSE/SCAN_LIST/SCAN/OPTION_PROFILE/TITLE (#PCDATA)	
	The option profile title specified for the scan.
/SCAN_LIST_OUTPUT/RESPONSE/SCAN_LIST/SCAN/OPTION_PROFILE/DEFAULT_FLAG (#PCDATA)	
	A flag that specifies whether the option profile was defined as the default option profile in the user account. A value of 1 is returned when this option profile is the default. A value of 0 is returned when this option profile is not the default.

SCAP Scan List Output

The SCAP scan list output is an XML report returned from the SCAP scan list API call. The DTD can be found at the following URL (when your account is located on US Platform 1):

```
https://qualysapi.qualys.com/api/2.0/fo/scan/
qscap_scan_list_output.dtd
```

DTD for SCAP Scan List Output

A recent DTD for the SCAP scan list output (qscap_scan_list_output.dtd) is shown below.

```
<!-- QUALYS QSCAP SCAN_LIST_OUTPUT DTD -->
<!ELEMENT SCAN_LIST_OUTPUT (REQUEST?,RESPONSE)>

<!ELEMENT REQUEST (DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?,
                   POST_DATA?)>
<!ELEMENT DATETIME (#PCDATA)>
<!ELEMENT USER_LOGIN (#PCDATA)>
<!ELEMENT RESOURCE (#PCDATA)>
<!ELEMENT PARAM_LIST (PARAM+)>
<!ELEMENT PARAM (KEY, VALUE)>
<!ELEMENT KEY (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>
<!-- if returned, POST_DATA will be urlencoded -->
<!ELEMENT POST_DATA (#PCDATA)>

<!ELEMENT RESPONSE (DATETIME, SCAN_LIST?)>
<!ELEMENT SCAN_LIST (SCAN+)>
<!ELEMENT SCAN (ID?, REF, TYPE, TITLE, POLICY, USER_LOGIN,
                LAUNCH_DATETIME, STATUS?, TARGET, ASSET_GROUP_TITLE_LIST?,
                OPTION_PROFILE?)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT REF (#PCDATA)>
<!ELEMENT TYPE (#PCDATA)>
<!ELEMENT TITLE (#PCDATA)>
<!ELEMENT POLICY (ID, TITLE)>
<!ELEMENT LAUNCH_DATETIME (#PCDATA)>
<!ELEMENT STATUS (STATE, SUB_STATE?)>
<!ELEMENT STATE (#PCDATA)>
<!ELEMENT SUB_STATE (#PCDATA)>
<!ELEMENT TARGET (#PCDATA)>
<!ELEMENT ASSET_GROUP_TITLE_LIST (ASSET_GROUP_TITLE+)>
<!ELEMENT ASSET_GROUP_TITLE (#PCDATA)>
<!ELEMENT OPTION_PROFILE (TITLE, DEFAULT_FLAG?)>
<!ELEMENT DEFAULT_FLAG (#PCDATA)>
```

XPaths for SCAP Scan List Output

This section describes the XPaths for the SCAP scan list output (scap_scan_list_output.dtd).

XPath	element specifications / notes
/SCAN_LIST_OUTPUT	(REQUEST?, RESPONSE)
/SCAN_LIST_OUTPUT/REQUEST	(DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?, POST_DATA?)
/SCAN_LIST_OUTPUT/REQUEST/DATETIME	(#PCDATA) The date and time of the request.
/SCAN_LIST_OUTPUT/REQUEST/USER_LOGIN	(#PCDATA) The user login ID of the user who made the request.
/SCAN_LIST_OUTPUT/REQUEST/RESOURCE	(#PCDATA) The resource specified for the request.
/SCAN_LIST_OUTPUT/REQUEST/PARAM_LIST	(PARAM+)
/SCAN_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM	(KEY, VALUE)
/SCAN_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM/KEY	(#PCDATA) The input parameter name.
/SCAN_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM/VALUE	(#PCDATA) The input parameter value.
/SCAN_LIST_OUTPUT/REQUEST/POST_DATA	(#PCDATA) The POST data, if any.
/SCAN_LIST_OUTPUT/RESPONSE	(DATETIME, SCAN_LIST?)
/SCAN_LIST_OUTPUT/RESPONSE/SCAN_LIST	(SCAN+)
/SCAN_LIST_OUTPUT/RESPONSE/SCAN_LIST/SCAN	(ID?, REF, TYPE, TITLE, USER_LOGIN, LAUNCH_DATETIME, STATUS?, TARGET, ASSET_GROUP_TITLE_LIST?, OPTION_PROFILE?
/SCAN_LIST_OUTPUT/RESPONSE/SCAN_LIST/SCAN/ID	(#PCDATA) The SCAP scan ID.
/SCAN_LIST_OUTPUT/RESPONSE/SCAN_LIST/SCAN/REF	(#PCDATA) The SCAP scan reference code.
/SCAN_LIST_OUTPUT/RESPONSE/SCAN_LIST/SCAN/TYPE	(#PCDATA) The scan type: On-Demand, Scheduled or API.
/SCAN_LIST_OUTPUT/RESPONSE/SCAN_LIST/SCAN/TITLE	(#PCDATA) The SCAP scan title.
/SCAN_LIST_OUTPUT/RESPONSE/SCAN_LIST/SCAN/POLICY	(ID, TITLE)
/SCAN_LIST_OUTPUT/RESPONSE/SCAN_LIST/SCAN/POLICY/ID	(#PCDATA) The SCAP policy ID.

XPath	element specifications / notes
/SCAN_LIST_OUTPUT/RESPONSE/SCAN_LIST/SCAN/POLICY/TITLE (#PCDATA)	
	The SCAP policy title.
/SCAN_LIST_OUTPUT/RESPONSE/SCAN_LIST/SCAN/USER_LOGIN (#PCDATA)	
	The user login ID of the user who launched the SCAP scan.
/SCAN_LIST_OUTPUT/RESPONSE/SCAN_LIST/SCAN/LAUNCH_DATETIME (#PCDATA)	
	The date and time when the SCAP scan was launched.
/SCAN_LIST_OUTPUT/RESPONSE/SCAN_LIST/SCAN/STATUS	
	(STATE, SUB-STATE?)
/SCAN_LIST_OUTPUT/RESPONSE/SCAN_LIST/SCAN/STATUS/STATE (#PCDATA)	
	The scan state: Running, Paused, Canceled, Finished, Error, Queued (scan job is waiting to be distributed to scanner(s)), or Loading (scanner(s) are finished and scan results are being loaded onto the platform).
/SCAN_LIST_OUTPUT/RESPONSE/SCAN_LIST/SCAN/STATUS/SUB_STATE (#PCDATA)	
	The sub-state related to the scan state, if any. For scan state Finished, value can be: No_Vuln (no vulnerabilities found) or No_Host (no host alive). For scan state Queued, value can be: Launching (service received scan request), Pausing (service received pause scan request), or Resuming (service received resume scan request).
/SCAN_LIST_OUTPUT/RESPONSE/SCAN_LIST/SCAN/TARGET (#PCDATA)	
	The target hosts selected for the SCAP scan.
/SCAN_LIST_OUTPUT/RESPONSE/SCAN_LIST/SCAN/ASSET_GROUP_TITLE_LIST (ASSET_GROUP_TITLE+)	
/SCAN_LIST_OUTPUT/RESPONSE/SCAN_LIST/SCAN/ASSET_GROUP_TITLE_LIST/ASSET_GROUP_TITLE	
	(#PCDATA)
	The asset group title selected for the SCAP scan.
/SCAN_LIST_OUTPUT/RESPONSE/SCAN_LIST/SCAN/OPTION_PROFILE (TITLE, DEFAULT_FLAG?)	
/SCAN_LIST_OUTPUT/RESPONSE/SCAN_LIST/SCAN/OPTION_PROFILE/TITLE (#PCDATA)	
	The option profile title seleted for the SCAP scan.
/SCAN_LIST_OUTPUT/RESPONSE/SCAN_LIST/SCAN/OPTION_PROFILE/DEFAULT_FLAG (#PCDATA)	
	A flag that specifies whether the option profile was defined as the default option profile in the user account. A value of 1 is returned when this option profile is the default. A value of 0 is returned when this option profile is not the default.

Scheduled Scan List Output

The scheduled scan list output is an XML report returned from scheduled scan API calls. The DTD can be found at the following URL (when your account is located on US Platform 1):

```
https://qualysapi.qualys.com/api/2.0/fo/schedule/scan/schedule_
scan_list_output.dtd
```

DTD for Scheduled Scan List Output

A recent DTD for the scheduled scan list output (scheduled_scan_list_output.dtd) is shown below.

```
<!-- QUALYS SCHEDULE_SCAN_LIST_OUTPUT DTD -->

<!ELEMENT SCHEDULE_SCAN_LIST_OUTPUT (REQUEST?,RESPONSE)>

<!ELEMENT REQUEST (DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?,
POST_DATA?)>
<!ELEMENT DATETIME (#PCDATA)>
<!ELEMENT USER_LOGIN (#PCDATA)>
<!ELEMENT RESOURCE (#PCDATA)>
<!ELEMENT PARAM_LIST (PARAM+)>
<!ELEMENT PARAM (KEY, VALUE)>
<!ELEMENT KEY (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>
<!-- if returned, POST_DATA will be urlencoded -->
<!ELEMENT POST_DATA (#PCDATA)>

<!ELEMENT RESPONSE (DATETIME, SCHEDULE_SCAN_LIST?)>
<!ELEMENT SCHEDULE_SCAN_LIST (SCAN+)>
<!ELEMENT SCAN (ID, ACTIVE, TITLE?, USER_LOGIN, TARGET, NETWORK_ID?,
ISCANNER_NAME?, EC2_INSTANCE?, ASSET_GROUP_TITLE_LIST?, ASSET_TAGS?,
EXCLUDE_IP_PER_SCAN?, USER_ENTERED_IPS?, OPTION_PROFILE?, SCHEDULE)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT ACTIVE (#PCDATA)>
<!ELEMENT TITLE (#PCDATA)>
<!ELEMENT TARGET (#PCDATA)>
<!ELEMENT NETWORK_ID (#PCDATA)>
<!ELEMENT ISCANNER_NAME (#PCDATA)>
<!ELEMENT EC2_INSTANCE (CONNECTOR_UUID, EC2_ENDPOINT, EC2_ONLY_CLASSIC?)>
<!ELEMENT CONNECTOR_UUID (#PCDATA)>
<!ELEMENT EC2_ENDPOINT (#PCDATA)>
<!ELEMENT EC2_ONLY_CLASSIC (#PCDATA)>
<!ELEMENT ASSET_GROUP_TITLE_LIST (ASSET_GROUP_TITLE+)>
<!ELEMENT ASSET_GROUP_TITLE (#PCDATA)>
```

```
<!ELEMENT ASSET_TAGS (TAG_INCLUDE_SELECTOR, TAG_SET_INCLUDE,
TAG_EXCLUDE_SELECTOR?, TAG_SET_EXCLUDE?, USE_IP_NT_RANGE_TAGS)>
<!ELEMENT TAG_INCLUDE_SELECTOR (#PCDATA)>
<!ELEMENT TAG_SET_INCLUDE (#PCDATA)>
<!ELEMENT TAG_EXCLUDE_SELECTOR (#PCDATA)>
<!ELEMENT TAG_SET_EXCLUDE (#PCDATA)>
<!ELEMENT USE_IP_NT_RANGE_TAGS (#PCDATA)>

<!ELEMENT EXCLUDE_IP_PER_SCAN (#PCDATA)>
<!ELEMENT USER_ENTERED_IPS (RANGE+)>
<!ELEMENT RANGE (START, END)>
<!ELEMENT START (#PCDATA)>
<!ELEMENT END (#PCDATA)>
<!ELEMENT OPTION_PROFILE (TITLE, DEFAULT_FLAG?)>
<!ELEMENT DEFAULT_FLAG (#PCDATA)>

<!ELEMENT SCHEDULE ((DAILY|WEEKLY|MONTHLY), START_DATE_UTC, START_HOUR,
START_MINUTE, PAUSE_AFTER_HOURS?, RESUME_IN_DAYS?, NEXTLAUNCH_UTC?,
TIME_ZONE, DST_SELECTED, MAX_OCCURRENCE?)>
<!ELEMENT DAILY EMPTY>
<!ATTLIST DAILY
       frequency_days  CDATA #REQUIRED>

<!-- weekdays is comma-separated list of weekdays e.g. 0,1,4,5 -->
<!ELEMENT WEEKLY EMPTY>
<!ATTLIST WEEKLY
       frequency_weeks  CDATA #REQUIRED
       weekdays   CDATA #REQUIRED>

<!-- either day of month, or (day of week and week of month) must be
provided -->
<!ELEMENT MONTHLY EMPTY>
<!ATTLIST MONTHLY
       frequency_months  CDATA #REQUIRED
       day_of_month  CDATA  #IMPLIED
       day_of_week  (0|1|2|3|4|5|6)  #IMPLIED
       week_of_month  (1|2|3|4|5)  #IMPLIED>

<!-- start date of the task in UTC -->
<!ELEMENT START_DATE_UTC (#PCDATA)>
<!-- User Selected hour -->
<!ELEMENT START_HOUR (#PCDATA)>
<!-- User Selected Minute -->
<!ELEMENT START_MINUTE (#PCDATA)>
<!ELEMENT PAUSE_AFTER_HOURS (#PCDATA)>
<!ELEMENT RESUME_IN_DAYS (#PCDATA)>
<!ELEMENT NEXTLAUNCH_UTC (#PCDATA)>
<!ELEMENT TIME_ZONE (TIME_ZONE_CODE, TIME_ZONE_DETAILS)>
```

```
<!-- timezone code like US-CA -->
<!ELEMENT TIME_ZONE_CODE (#PCDATA)>

<!-- timezone details like (GMT-0800) United States (California): Los
Angeles, Sacramento, San Diego, San Francisco-->
<!ELEMENT TIME_ZONE_DETAILS (#PCDATA)>

<!-- Did user select DST? 0-not selected 1-selected -->
<!ELEMENT DST_SELECTED (#PCDATA)>
<!ELEMENT MAX_OCCURRENCE (#PCDATA)>
```

XPaths for Scheduled Scan List Output

This section describes the XPaths for the scheduled scan list output (scheduled_scan_list_output.dtd).

XPath	element specifications / notes
/SCHEDULE_SCAN_LIST_OUTPUT	(REQUEST?, RESPONSE)
/SCHEDULE_SCAN_LIST_OUTPUT/REQUEST	(DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?, POST_DATA?)
/SCHEDULE_SCAN_LIST_OUTPUT/REQUEST/DATETIME	(#PCDATA) The date and time of the request.
/SCHEDULE_SCAN_LIST_OUTPUT/REQUEST/USER_LOGIN	(#PCDATA) The user login ID of the user who made the request.
/SCHEDULE_SCAN_LIST_OUTPUT/REQUEST/RESOURCE	(#PCDATA) The resource specified for the request.
/SCHEDULE_SCAN_LIST_OUTPUT/REQUEST/PARAM_LIST	(PARAM+)
/SCHEDULE_SCAN_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM	(KEY, VALUE)
/SCHEDULE_SCAN_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM/KEY	(#PCDATA) The input parameter name.
/SCHEDULE_SCAN_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM/VALUE	(#PCDATA) The input parameter value.
/SCHEDULE_SCAN_LIST_OUTPUT/REQUEST/POST_DATA	(#PCDATA) The POST data, if any.
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE	(DATETIME, SCHEDULE_SCAN_LIST?)
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST	(SCAN+)

XPath	element specifications / notes
/SCHEDULE_CAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN	(ID, ACTIVE, TITLE?, USER_LOGIN, TARGET, NETWORK_ID?, ISCANNER_NAME?, EC2_INSTANCE?, ASSET_GROUP_TITLE_LIST?, ASSET_TAGS?, EXCLUDE_IP_PER_SCAN?, USER_ENTERED_IPS?, OPTION_PROFILE?, SCHEDULE)
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/ID (#PCDATA)	The scan ID.
/SSCHEDULE_CAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/ACTIVE (#PCDATA)	1 for an active schedule, or 0 for a deactivated schedule.
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/TITLE (#PCDATA)	The scan title.
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/USER_LOGIN (#PCDATA)	The user login ID for the user who owns the scan schedule.
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/TARGET (#PCDATA)	The target hosts for the scan.
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/NETWORK_ID (#PCDATA)	The network ID for the target hosts, if custom networks are defined.
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/ ISCANNER_NAME (#PCDATA)	The name of the scanner appliance used for the scan.
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/EC2_INSTANCE	(CONNECTOR_UUID, EC2_ENDPOINT, EC2_ONLY_CLASSIC?)
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/EC2_INSTANCE/ CONNECTOR_UUID (#PCDATA)	The connector uuid for the AWS integration used for the EC2 scan.
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/EC2_INSTANCE/ EC2_ENDPOINT (#PCDATA)	The EC2 region code, or the ID of the Virtual Private Cloud (VPC) zone.
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/EC2_INSTANCE/ EC2_ONLY_CLASSIC (#PCDATA)	1 means the EC2 scan is configured to scan EC2 classic hosts in the region.
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/ASSET_GROUP_TITLE_LIST (ASSET_GROUP_TITLE+)	
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/ASSET_GROUP_TITLE_LIST/ASSET_GROUP_TITLE (#PCDATA)	The asset group title specified for the scan.
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/ASSET_TAGS	(TAG_INCLUDE_SELECTOR, TAG_SET_INCLUDE, TAG_EXCLUDE_SELECTOR?, TAG_SET_EXCLUDE?, USE_IP_NT_RANGE_TAGS)

XPath	element specifications / notes
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/ASSET_TAGS/TAG_INCLUDE_SELECTOR (#PCDATA)	
	Include any of the selected tags (any) or all of the selected tags (all).
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/ASSET_TAGS/TAG_SET_INCLUDE (#PCDATA)	
	Tag set to include from the scan target.
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/ASSET_TAGS/TAG_EXCLUDE_SELECTOR (#PCDATA)	
	Exclude any of the selected tags (any) or all of the selected tags (all).
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/ASSET_TAGS/TAG_SET_EXCLUDE (#PCDATA)	
	Tag set to exclude from the scan target.
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/ASSET_TAGS/USE_IP_NT_RANGE_TAGS (#PCDATA)	
	0 means select from all tags (tags with any tag rule). 1 means scan all IP addresses defined in tags with the rule "IP address in Network Range(s)".
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/EXCLUDE_IP_PER_SCAN (#PCDATA)	
	When the scan target has excluded hosts, the target hosts that were excluded.
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/USER_ENTERED_IPS (RANGE+)	
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/RANGE (START, END)	
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/RANGE/START (#PCDATA)	
	When the scan target includes user entered IPs, the start of an IP range.
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/RANGE/END (#PCDATA)	
	When the scan target includes user entered IPs, the end of an IP range.
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/OPTION_PROFILE (TITLE, DEFAULT_FLAG?)	
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/OPTION_PROFILE/TITLE (#PCDATA)	
	The option profile title specified for the scan.
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/OPTION_PROFILE/DEFAULT_FLAG (#PCDATA)	
	A flag that specifies whether the option profile was defined as the default option profile in the user account. A value of 1 is returned when this option profile is the default. A value of 0 is returned when this option profile is not the default.
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/SCHEDULE ((DAILY|WEEKLY|MONTHLY), START_DATE_UTC, START_HOUR, START_MINUTE, PAUSE_AFTER_HOURS?, RESUME_IN_DAYS?, NEXTLAUNCH_UTC?, TIME_ZONE, DST_SELECTED, MAX_OCCURRENCE?)	
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/SCHEDULE /DAILY	
attribute: **frequency_days**	**frequency_days** is *required* for a scan that runs after some number of days (from 1 to 365)

XPath	element specifications / notes
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/SCHEDULE /WEEKLY	
attribute: **frequency_weeks**	**frequency_weeks** is *required* for a scan that runs after some number of weeks (from 1 to 52)
attribute: **weekdays**	**weekdays** is *required* for a scan that runs after some number of weeks on a particular weekday (from 0 to 6), where 0 is Sunday and 6 is Saturday, multiple weekdays are comma separated
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/SCHEDULE /MONTHLY	
attribute: **frequency_months**	**frequency_months** is *required* for a scan that runs after some number of months (from 1 to 12)
attribute: **day_of_month**	**day_of_month** is *implied* and, if present, indicates the scan runs on the Nth day of the month (from 1 to 31)
attribute: **day_of_week**	**day_of_week** is *implied* and, if present, indicates the scan runs on the Nth day of the month on a particular weekday (from 0 to 6), where 0 is Sunday and 6 is Saturday
attribute: **week_of_month**	**week_of_month** is *implied* and, if present, indicates the scan runs on the Nth day of the month on the Nth week of the month (from 1 to 5), where 1 is the first week of the month and 5 is the fifth week of the month
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/SCHEDULE / START_DATE_UTC (#PCDATA)	
	The start date (in UTC format) defined for the scan schedule.
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/SCHEDULE / START_HOUR (#PCDATA)	
	The start hour defined for the scan schedule.
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/SCHEDULE /START_MINUTE (#PCDATA)	
	The start minute defined for the scan schedule.
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/SCHEDULE / PAUSE_AFTER_HOURS (#PCDATA)	
	The “pause after number of hours” setting defined for the scan schedule.
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/SCHEDULE / RESUME_IN_DAYS (#PCDATA)	
	The “resume in number of days” setting defined for the scan schedule.
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/SCHEDULE / NEXTLAUNCH_UTC (#PCDATA)	
	The next launch date and time for the scan schedule.
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/SCHEDULE / SCHEDULE/TIME_ZONE (TIME_ZONE_CODE, TIME_ZONE_DETAILS)	
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/SCHEDULE / TIME_ZONE/TIME_ZONE_CODE (#PCDATA)	
	The time zone code defined for the scan schedule. For example: US-CA.

XPath	element specifications / notes
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/SCHEDULE / TIME_ZONE/TIME_ZONE_DETAILS (#PCDATA)	
	The time zone details (description) for the local time zone, identified in the <TIME_ZONE_CODE> element. For example:, (GMT-0800) United States (California): Los Angeles, Sacramento, San Diego, San Francisco.
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/SCHEDULE / DST_SELECTED (#PCDATA)	
	When set to 1, Daylight Saving Time (DST) is enabled for the scan schedule.
/SCHEDULE_SCAN_LIST_OUTPUT/RESPONSE/SCHEDULE_SCAN_LIST/SCAN/SCHEDULE / MAX_OCCURRENCE (#PCDATA)	
	The number of times the scan schedule will be run before it is deactivated (from 1 to 99).

Vulnerability Scan Results

The vulnerability scan results is returned from the download vulnerability scan results API call. Vulnerability scan results can be downloaded in these formats: CSV and JSON (JavaScript Object Notation). The scan results will be in brief or extended format.

This information is returned in both brief and extended formats.

Field	Description
IP	IP address.
DNS Name	DNS hostname when available.
Netbios Name	NetBIOS hostname when available.
QID	Qualys vulnerability ID (QID).
Result	Scan test result returned by the scanning engine.

This additional information is returned in extended format only.

Field	Description
Protocol	Protocol used to detect the vulnerability.
Port	Port used to detect the vulnerability.
SSL	A flag indicating whether SSL was used to detect the vulnerability: “yes” indicates SSL was used to detect the vulnerability, “no” indicates SSL was not used to detect the vulnerability.
FQDN	Fully qualified domain name for the host, when defined.

Compliance Scan Results

The compliance scan results output XML is returned from an API request to fetch (download) a compliance scan using the PC scan API (/api/2.0/fo/scan/compliance/?action=fetch). The DTD can be found at the following URL (when your account is located on US Platform 1):

```
https://qualysapi.qualys.com/api/2.0/fo/scan/compliance/compliance_scan_result_output.dtd
```

DTD for Compliance Scan Result Output

The PC scan results output DTD (compliance_scan_result_output.dtd) is below.

```
<!ELEMENT COMPLIANCE_SCAN_RESULT_OUTPUT (REQUEST?,RESPONSE)>

<!ELEMENT REQUEST (DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?,
                   POST_DATA?)>
<!ELEMENT DATETIME (#PCDATA)>
<!ELEMENT USER_LOGIN (#PCDATA)>
<!ELEMENT RESOURCE (#PCDATA)>
<!ELEMENT PARAM_LIST (PARAM+)>
<!ELEMENT PARAM (KEY, VALUE)>
<!ELEMENT KEY (#PCDATA)>
<!ATTLIST KEY
    value CDATA #IMPLIED
>
<!ELEMENT VALUE (#PCDATA)>
<!-- if returned, POST_DATA will be urlencoded -->
<!ELEMENT POST_DATA (#PCDATA)>

<!ELEMENT RESPONSE (DATETIME, COMPLIANCE_SCAN)>
<!ELEMENT COMPLIANCE_SCAN ((HEADER, ERROR?, AUTH_SCAN_ISSUES?,
                            APPENDIX)+)>
<!ELEMENT ERROR (#PCDATA)>
<!ATTLIST ERROR
    number CDATA #IMPLIED
>
<!-- INFORMATION ABOUT THE SCAN -->
<!ELEMENT HEADER (NAME, GENERATION_DATETIME, COMPANY_INFO, USER_INFO,
                  KEY+, ASSET_GROUPS?, OPTION_PROFILE?)>
<!ELEMENT NAME (#PCDATA)*>
<!ELEMENT GENERATION_DATETIME (#PCDATA)*>

<!ELEMENT COMPANY_INFO (NAME, ADDRESS, CITY, STATE, COUNTRY, ZIP_CODE)>
<!ELEMENT ADDRESS (#PCDATA)>
<!ELEMENT CITY (#PCDATA)>
```

```
<!ELEMENT STATE (#PCDATA)>
<!ELEMENT COUNTRY (#PCDATA)>
<!ELEMENT ZIP_CODE (#PCDATA)>

<!ELEMENT USER_INFO (NAME, USERNAME, ROLE)>
<!ELEMENT USERNAME (#PCDATA)*>
<!ELEMENT ROLE (#PCDATA)*>

<!-- NAME of the asset group with the TYPE attribute with possible values
of (DEFAULT | EXTERNAL | ISCANNER) -->
<!ELEMENT ASSET_GROUP (ASSET_GROUP_TITLE)>
<!ELEMENT ASSET_GROUPS (ASSET_GROUP+)>
<!ELEMENT ASSET_GROUP_TITLE (#PCDATA)>
<!ELEMENT OPTION_PROFILE (OPTION_PROFILE_TITLE)>
<!ELEMENT OPTION_PROFILE_TITLE (#PCDATA)>
<!ATTLIST OPTION_PROFILE_TITLE
    option_profile_default CDATA #IMPLIED
>
<!ELEMENT AUTH_SCAN_ISSUES (AUTH_SCAN_FAILED*, AUTH_SCAN_INSUFFICIENT*)>
<!ELEMENT AUTH_SCAN_FAILED (HOST_INFO*)>
<!ELEMENT AUTH_SCAN_INSUFFICIENT (HOST_INFO*)>
<!ELEMENT HOST_INFO (DNS, IP, NETBIOS, INSTANCE, CAUSE)>
<!ELEMENT DNS (#PCDATA)>
<!ELEMENT IP (#PCDATA)>
<!ELEMENT NETBIOS (#PCDATA)>
<!ELEMENT INSTANCE (#PCDATA)>
<!ELEMENT CAUSE (#PCDATA)>

<!ELEMENT APPENDIX (TARGET_HOSTS?, TARGET_DISTRIBUTION?,
                    AUTHENTICATION?)>
<!ELEMENT TARGET_HOSTS (HOSTS_SCANNED?, EXCLUDED_HOSTS?,
                        HOSTS_NOT_ALIVE?, PAUSE_CANCEL_ACTION?,
                        HOSTNAME_NOT_FOUND?,  HOSTS_SCAN_ABORTED?)>
<!ELEMENT HOSTS_SCANNED (#PCDATA)>
<!ELEMENT HOSTNAME_NOT_FOUND (#PCDATA)>
<!ELEMENT EXCLUDED_HOSTS (#PCDATA)>
<!ELEMENT HOSTS_NOT_ALIVE (#PCDATA)>
<!ELEMENT HOSTS_SCAN_ABORTED (#PCDATA)>
<!ELEMENT PAUSE_CANCEL_ACTION (HOSTS, ACTION, BY)>
<!ELEMENT ACTION (#PCDATA)>
<!ELEMENT BY (#PCDATA)>

<!ELEMENT TARGET_DISTRIBUTION (SCANNER+)>
<!ELEMENT SCANNER (NAME, HOSTS)>
<!ELEMENT HOSTS (#PCDATA)>

<!ELEMENT AUTHENTICATION (AUTH+)>
<!ELEMENT AUTH (TYPE?, (FAILED | SUCCESS | INSUFFICIENT)+)>
<!ELEMENT TYPE (#PCDATA)>
```

```
<!ELEMENT FAILED (IP,INSTANCE?)>
<!ELEMENT SUCCESS (IP,INSTANCE?)>
<!ELEMENT INSUFFICIENT (IP,INSTANCE?)>
<!-- EOF -->
```

XPaths for Compliance Scan Result Output

This section describes the XPaths for the compliance scan results output.

XPath	element specifications / notes
/COMPLIANCE_SCAN_RESULT_OUTPUT (REQUEST?, RESPONSE)	
/COMPLIANCE_SCAN_RESULT_OUTPUT/REQUEST	(DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?, POST_DATA)
/COMPLIANCE_SCAN_RESULT_OUTPUT/REQUEST/DATETIME (#PCDATA)	The date and time the scan was launched.
/COMPLIANCE_SCAN_RESULT_OUTPUT/REQUEST/USER_LOGIN (#PCDATA)	The login ID of the user who launched the scan.
/COMPLIANCE_SCAN_RESULT_OUTPUT/REQUEST/RESOURCE (#PCDATA)	The resource specified for the request.
/COMPLIANCE_SCAN_RESULT_OUTPUT/REQUEST/PARAM_LIST (PARAM+)	
/COMPLIANCE_SCAN_RESULT_OUTPUT/REQUEST/PARAM_LIST/PARAM (KEY, VALUE)	
/COMPLIANCE_SCAN_RESULT_OUTPUT/REQUEST/PARAM_LIST/PARAM/KEY (#PCDATA)	An input parameter name.
/COMPLIANCE_SCAN_RESULT_OUTPUT/REQUEST/PARAM_LIST/PARAM/VALUE (#PCDATA)	An input parameter value.
/COMPLIANCE_SCAN_RESULT_OUTPUT/REQUEST/POST_DATA (#PCDATA)	The POST data.
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE (DATETIME, COMPLIANCE_SCAN)	
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/DATETIME (#PCDATA)	The date and time of the response.
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN	((HEADER, ERROR?, AUTH_SCAN_ISSUES?, APPENDIX)+)
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/ HEADER	(NAME, GENERATION_DATETIME, COMPANY_INFO, USER_INFO, KEY+ ASSET_GROUPS?, OPTION PROFILE?)

XPath	element specifications / notes
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/ HEADER/NAME (#PCDATA)	
	The name of the scan.
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/ HEADER/GENERATION_DATETIME (#PCDATA)	
	The date and time when the scan was launched.
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/ HEADER/COMPANY_INFO	
	(NAME, ADDRESS, CITY, STATE, COUNTRY, ZIP_CODE)
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/ HEADER/COMPANY_INFO/NAME (#PCDATA)	
	The company name associated with the account used to launch the scan.
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/ HEADER/COMPANY_INFO/ADDRESS (#PCDATA)	
	The street address associated with the account used to launch the scan.
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/ HEADER/COMPANY_INFO/CITY (#PCDATA)	
	The city associated with the account used to launch the scan.
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/ HEADER/COMPANY_INFO/STATE (#PCDATA)	
	The city associated with the account used to launch the scan.
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/ HEADER/COMPANY_INFO/COUNTRY (#PCDATA)	
	The country associated with the account used to launch the scan.
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/ HEADER/COMPANY_INFO/ZIP_CODE (#PCDATA)	
	The zip code associated with the account used to launch the scan.
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/ HEADER/USER_INFO	
	(NAME, USERNAME, ROLE)
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/ HEADER/USER_INFO/NAME (#PCDATA)	
	The name of the user who launched the scan.
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/ HEADER/USER_INFO/USERNAME (#PCDATA)	
	The user login of the user who launched the scan.
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/ HEADER/USER_INFO/ROLE (#PCDATA)	
	The user role assigned to the user who launched the scan.
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/ HEADER/ASSET_GROUPS/ASSET_GROUP	
	(ASSET_GROUP_TITLE)

XPath	element specifications / notes
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/ HEADER/ASSET_GROUPS (ASSET_GROUP+)	
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/ HEADER/ASSET_GROUPS /ASSET_GROUP_TITLE (#PCDATA)	
	The title of an asset group in the scan target.
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/ HEADER/OPTION_PROFILE	
	(OPTION_PROFILE_TITLE)
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/ HEADER/OPTION_PROFILE/OPTION_PROFILE_TITLE (#PCDATA)	
	The title of the option profile used.
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/ERROR (#PCDATA)	
	An error description.
attribute: number	An error number (implied)
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/AUTH_SCAN_ISSUES	
	(AUTH_SCAN_FAILED, AUTH_SCAN_INSUFFICIENT)
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/ AUTH_SCAN_ISSUES/AUTH_SCAN_FAILED (HOST_INFO)	
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/ AUTH_SCAN_ISSUES/AUTH_SCAN_FAILED/HOST_INFO (DNS, IP, NETBIOS, INSTANCE, CAUSE)	
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/ AUTH_SCAN_ISSUES/AUTH_SCAN_FAILED/HOST_INFO/DNS (#PCDATA)	
	The DNS name of a host that failed authentication.
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/ AUTH_SCAN_ISSUES/AUTH_SCAN_FAILED/HOST_INFO/IP (#PCDATA)	
	The IP address of a host that failed authentication.
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/ AUTH_SCAN_ISSUES/AUTH_SCAN_FAILED/HOST_INFO/NETBIOS (#PCDATA)	
	The NetBIOS hostname of a host that failed authentication.
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/ AUTH_SCAN_ISSUES/AUTH_SCAN_FAILED/HOST_INFO/INSTANCE (#PCDATA)	
	The instance of a host that failed authentication.
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/ AUTH_SCAN_ISSUES/AUTH_SCAN_FAILED/HOST_INFO/CAUSE (#PCDATA)	
	Additional information for a host that failed authentication. This may include the login ID used during the authentication attempt.
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/ AUTH_SCAN_ISSUES/AUTH_SCAN_INSUFFICIENT (HOST_INFO)	
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/ AUTH_SCAN_ISSUES/AUTH_SCAN_INSUFFICIENT/HOST_INFO	
	(DNS, IP, NETBIOS, INSTANCE, CAUSE)

XPath	element specifications / notes
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/AUTH_SCAN_ISSUES/AUTH_SCAN_INSUFFICIENT/HOST_INFO/DNS (#PCDATA)	
	The DNS name of a host that failed authentication due to insufficient privileges.
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/AUTH_SCAN_ISSUES/AUTH_SCAN_INSUFFICIENT/HOST_INFO/IP (#PCDATA)	
	The IP address of a host that failed authentication due to insufficient privileges.
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/AUTH_SCAN_ISSUES/AUTH_SCAN_INSUFFICIENT/HOST_INFO/NETBIOS (#PCDATA)	
	The NetBIOS hostname of a host that failed authentication due to insufficient privileges.
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/AUTH_SCAN_ISSUES/AUTH_SCAN_INSUFFICIENT/HOST_INFO/INSTANCE (#PCDATA)	
	The instanceof the host that failed authentication due to insufficient privileges..
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/AUTH_SCAN_ISSUES/AUTH_SCAN_INSUFFICIENT/HOST_INFO/CAUSE (#PCDATA)	
	Additional information for a host that failed authentication due to insufficient privileges. This may include the login ID used during the authentication attempt.
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/APPENDIX	
	(TARGET_HOSTS,?, TARGET DISTRIBUTION?, AUTHENTICATION?)
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/APPENDIX/TARGET_HOSTS	
	(HOSTS_SCANNED?, EXCLUDED_HOSTS?, HOSTS_NOT_ALIVE?, PAUSE_CANCEL_ACTION?, HOSTNAME_NOT_FOUND?, HOSTS_SCAN_ABORTED?)
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/APPENDIX/TARGET_HOSTS/HOSTS_SCANNED (#PCDATA)	
	Target hosts that were scanned.
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/APPENDIX/TARGET_HOSTS/EXCLUDED_HOSTS (#PCDATA)	
	Target hosts that were excluded from the scan target.
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/APPENDIX/TARGET_HOSTS/HOSTS_NOT_ALIVE (#PCDATA)	
	Target hosts that were not alive.
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/APPENDIX/TARGET_HOSTS/HOSTNAME_NOT_FOUND (#PCDATA)	
	Target hosts that were not found.
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/APPENDIX/TARGET_HOSTS/HOSTS_SCAN_ABORTED (#PCDATA)	
	Target hosts on which the scan was aborted.
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/APPENDIX/TARGET_HOSTS/PAUSE_CANCEL_ACTION (HOSTS, ACTION, BY)	

XPath	element specifications / notes
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/APPENDIX/TARGET_HOSTS/PAUSE_CANCEL_ACTION/HOSTS (#PCDATA)	
	The target hosts that an action (pause or cancel) was taken on.
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/APPENDIX/TARGET_HOSTS/PAUSE_CANCEL_ACTION/ACTION (#PCDATA)	
	An action (pause or cancel) taken by a user on a scan.
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/APPENDIX/TARGET_HOSTS/PAUSE_CANCEL_ACTION/BY (#PCDATA)	
	The user who took an action (pause or cancel).
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/APPENDIX/TARGET_DISTRIBUTION (SCANNER+)	
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/APPENDIX/TARGET_DISTRIBUTION/SCANNER (NAME, HOSTS)	
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/APPENDIX/TARGET_DISTRIBUTION/SCANNER/NAME (#PCDATA)	
	The name of a scanner appliance used.
/COMPLIANCE_SCAN_RESULT_OUTPUT/RESPONSE/COMPLIANCE_SCAN/APPENDIX/TARGET_DISTRIBUTION/SCANNER/HOSTS (#PCDATA)	
	The compliance hosts that were scanned.

PCI Scan Share Status Output

The PCI scan share status output is an XML report returned from an API request for the share status of a PCI scan that's already been shared with a PCI Merchant account.

The DTD "pci_scan_share_status.dtd" can be found at the following URL:

```
https://qualysapi.qualys.com/api/2.0/fo/scan/pci/
pci_scan_share_status.dtd
```

where <qualysapi.qualys.com> is the API server URL where your account is located.

DTD for PCI Scan Share Status Output

A recent DTD for the PCI scan share status output (pci_scan_share_status.dtd) is shown below.

```
<!-- QUALYS PCI_SCAN_SHARE_STATUS DTD -->

<!ELEMENT PCI_SCAN_SHARE_STATUS (REQUEST?,RESPONSE)>

<!ELEMENT REQUEST (DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?,
                   POST_DATA?)>
<!ELEMENT DATETIME (#PCDATA)>
<!ELEMENT USER_LOGIN (#PCDATA)>
<!ELEMENT RESOURCE (#PCDATA)>
<!ELEMENT PARAM_LIST (PARAM+)>
<!ELEMENT PARAM (KEY, VALUE)>
<!ELEMENT KEY (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>
<!-- if returned, POST_DATA will be urlencoded -->
<!ELEMENT POST_DATA (#PCDATA)>

<!ELEMENT RESPONSE (SCAN)>
<!ELEMENT SCAN (MERCHANT_USERNAME, SCAN_REF, STATUS, LAST_SHARED)>
<!ELEMENT MERCHANT_USERNAME (#PCDATA)>
<!ELEMENT SCAN_REF (#PCDATA)>
<!ELEMENT LAST_SHARED (#PCDATA)>
<!ELEMENT STATUS (#PCDATA)>
<!-- EOF -->
```

XPaths for PCI Scan Share Status Output

This section describes the XPaths for the PCI scan share status output (pci_scan_share_status.dtd).

XPath	element specifications / notes
/PCI_SCAN_SHARE_STATUS	(REQUEST?, RESPONSE)
/PCI_SCAN_SHARE_STATUS/REQUEST	(DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?, POST_DATA?)
/PCI_SCAN_SHARE_STATUS/REQUEST/DATETIME	(#PCDATA) The date and time of the request.
/PCI_SCAN_SHARE_STATUS/REQUEST/USER_LOGIN	(#PCDATA) The user login ID of the user who made the request.
/PCI_SCAN_SHARE_STATUS/REQUEST/RESOURCE	(#PCDATA) The resource specified for the request.
/PCI_SCAN_SHARE_STATUS/REQUEST/PARAM_LIST	(PARAM+)
/PCI_SCAN_SHARE_STATUS/REQUEST/PARAM_LIST/PARAM	(KEY, VALUE)
/PCI_SCAN_SHARE_STATUS/REQUEST/PARAM_LIST/PARAM/KEY	(#PCDATA) The input parameter name.
/PCI_SCAN_SHARE_STATUS/REQUEST/PARAM_LIST/PARAM/VALUE	(#PCDATA) The input parameter value.
/PCI_SCAN_SHARE_STATUS/REQUEST/POST_DATA	(#PCDATA) The POST data, if any.
/PCI_SCAN_SHARE_STATUS/RESPONSE	(SCAN)
/PCI_SCAN_SHARE_STATUS/RESPONSE/SCAN	(MERCHANT_USERNAME, SCAN_REF, STATUS, LAST_SHARED)
/PCI_SCAN_SHARE_STATUS/RESPONSE/SCAN/MERCHANT_USERNAME	(#PCDATA) The user name for a target PCI Merchant account. This account is associated with a share PCI scan request.
/PCI_SCAN_SHARE_STATUS/RESPONSE/SCAN/SCAN_REF	(#PCDATA) The scan reference ID for the PCI scan associated. This PCI scan is associated with a share PCI scan request.
/PCI_SCAN_SHARE_STATUS/RESPONSE/SCAN/STATUS	(#PCDATA) The share status of a share PCI scan request for a PCI Merchant account and a PCI scan: Queued (request was received and sharing has not started yet), In Progress, Finished (request was successful and the scan was shared/exported to the PCI Merchant account successfully), or Error (request was not successful and the scan was not shared/exported).
/PCI_SCAN_SHARE_STATUS/RESPONSE/SCAN/LAST_SHARED	(#PCDATA) The most recent date and time of a share PCI scan request for a PCI Merchant account and a PCI scan.

Scanner Appliance List Output

The Scanner Appliance List API v2 (/api/2.0/fo/appliance/ with action=list) returns a list of scanner appliances using the "appliance _list_output.dtd". This DTD can be found at the following URL (where qualysapi.qualys.com is the base URL of the user's Qualys platform):

```
https://qualysapi.qualys.com/api/2.0/fo/appliance/
appliance_list_output.dtd
```

DTD for Scanner Appliance List Output

A recent DTD for the scanner appliance list output (appliance_list_output.dtd) is shown below.

```
<!-- QUALYS APPLIANCE_LIST_OUTPUT DTD -->
<!ELEMENT APPLIANCE_LIST_OUTPUT (REQUEST?,RESPONSE)>
<!ELEMENT REQUEST (DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?,
                   POST_DATA?)>
<!ELEMENT DATETIME (#PCDATA)>
<!ELEMENT USER_LOGIN (#PCDATA)>
<!ELEMENT RESOURCE (#PCDATA)>
<!ELEMENT PARAM_LIST (PARAM+)>
<!ELEMENT PARAM (KEY, VALUE)>
<!ELEMENT KEY (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>
<!-- if returned, POST_DATA will be urlencoded -->
<!ELEMENT POST_DATA (#PCDATA)>

<!ELEMENT RESPONSE (DATETIME, APPLIANCE_LIST?, LICENSE_INFO?)>
<!ELEMENT APPLIANCE_LIST (APPLIANCE+)>
<!ELEMENT APPLIANCE (ID, UUID, NAME, NETWORK_ID?, SOFTWARE_VERSION,
                     RUNNING_SCAN_COUNT, STATUS, MODEL_NUMBER?,
                     SERIAL_NUMBER?, ACTIVATION_CODE?,
                     INTERFACE_SETTINGS*, PROXY_SETTINGS?, VLANS?,
                     STATIC_ROUTES?, ML_LATEST?, ML_VERSION?,
                     VULNSIGS_LATEST?, VULNSIGS_VERSION?,
                     ASSET_GROUP_COUNT?, ASSET_GROUP_LIST?,
                     LAST_UPDATED_DATE?, POLLING_INTERVAL?, USER_LOGIN?,
                     HEARTBEATS_MISSED?, SS_CONNECTION?,
                     SS_LAST_CONNECTED?, FDCC_ENABLED?, USER_LIST?,
                     UPDATED?, COMMENTS?, RUNNING_SCANS?,
                     MAX_CAPACITY_UNITS?)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT UUID (#PCDATA)>
<!ELEMENT NAME (#PCDATA)>
<!ELEMENT NETWORK_ID (#PCDATA)>
```

```
<!ELEMENT SOFTWARE_VERSION (#PCDATA)>
<!ELEMENT RUNNING_SCAN_COUNT (#PCDATA)>
<!ELEMENT STATUS (#PCDATA)>
<!ELEMENT MODEL_NUMBER (#PCDATA)>
<!ELEMENT SERIAL_NUMBER (#PCDATA)>
<!ELEMENT ACTIVATION_CODE (#PCDATA)>
<!ELEMENT INTERFACE_SETTINGS (SETTING?, INTERFACE, IP_ADDRESS, NETMASK,
                              GATEWAY, LEASE, IPV6_ADDRESS?, SPEED,
                              DUPLEX, DNS)>
<!ELEMENT SETTING (#PCDATA)>
<!ELEMENT INTERFACE (#PCDATA)>
<!ELEMENT IP_ADDRESS (#PCDATA)>
<!ELEMENT NETMASK (#PCDATA)>
<!ELEMENT GATEWAY (#PCDATA)>
<!ELEMENT LEASE (#PCDATA)>
<!ELEMENT IPV6_ADDRESS (#PCDATA)>
<!ELEMENT SPEED (#PCDATA)>
<!ELEMENT DUPLEX (#PCDATA)>
<!ELEMENT DNS (DOMAIN?, PRIMARY, SECONDARY)>
<!ELEMENT DOMAIN (#PCDATA)>
<!ELEMENT PRIMARY (#PCDATA)>
<!ELEMENT SECONDARY (#PCDATA)>
<!ELEMENT PROXY_SETTINGS (SETTING, PROXY*)>
<!ELEMENT PROXY (IP_ADDRESS, PORT, USER)>
<!ELEMENT TYPE (#PCDATA)>
<!ELEMENT PORT (#PCDATA)>
<!ELEMENT USER (#PCDATA)>
<!ELEMENT VLANS (SETTING, VLAN*)>
<!ELEMENT STATIC_ROUTES (ROUTE*)>
<!ELEMENT ROUTE (NAME, IP_ADDRESS, NETMASK, GATEWAY)>
<!ELEMENT VLAN (ID, NAME, IP_ADDRESS, NETMASK)>
<!ELEMENT ML_LATEST (#PCDATA)>
<!ELEMENT ML_VERSION (#PCDATA)>
<!ATTLIST ML_VERSION updated CDATA #IMPLIED>
<!ELEMENT VULNSIGS_LATEST (#PCDATA)>
<!ELEMENT VULNSIGS_VERSION (#PCDATA)>
<!ATTLIST VULNSIGS_VERSION updated CDATA #IMPLIED>
<!ELEMENT ASSET_GROUP_COUNT (#PCDATA)>
<!ELEMENT ASSET_GROUP_LIST (ASSET_GROUP*)>
<!ELEMENT ASSET_GROUP (ID, NAME)>
<!ELEMENT LAST_UPDATED_DATE (#PCDATA)>
<!ELEMENT POLLING_INTERVAL (#PCDATA)>
<!ELEMENT HEARTBEATS_MISSED (#PCDATA)>
<!ELEMENT SS_CONNECTION (#PCDATA)>
<!ELEMENT SS_LAST_CONNECTED (#PCDATA)>
<!ELEMENT FDCC_ENABLED (#PCDATA)>
<!ELEMENT USER_LIST (USER_ACCOUNT*)>
<!ELEMENT USER_ACCOUNT (ID, NAME)>
<!ELEMENT UPDATED (#PCDATA)>
```

```
<!ELEMENT COMMENTS (#PCDATA)>
<!ELEMENT RUNNING_SCANS (SCAN+)>
<!ELEMENT MAX_CAPACITY_UNITS (#PCDATA)>
<!ELEMENT SCAN (ID, TITLE, REF, TYPE, SCAN_DATE)>
<!ELEMENT TITLE (#PCDATA)>
<!ELEMENT REF (#PCDATA)>
<!ELEMENT SCAN_DATE (#PCDATA)>

<!ELEMENT LICENSE_INFO (QVSA_LICENSES_COUNT, QVSA_LICENSES_USED)>
<!ELEMENT QVSA_LICENSES_COUNT (#PCDATA)>
<!ELEMENT QVSA_LICENSES_USED (#PCDATA)>
```

XPaths for Scanner Appliance List Output

This section describes the XPaths for the scanner appliance list output.

XPath	element specifications / notes
/APPLIANCE_LIST_OUTPUT	(REQUEST?,RESPONSE)
/APPLIANCE_LIST_OUTPUT/REQUEST	(DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?, POST_DATA?)
/APPLIANCE_LIST_OUTPUT/REQUEST/DATETIME	(#PCDATA)
	The date and time of the API request. (This element appears only when the API request includes the parameter **echo_request=1**.)
/APPLIANCE_LIST_OUTPUT/REQUEST/USER_LOGIN	(#PCDATA)
	The user login ID of the user who made the request. (This element appears only when the API request includes the parameter **echo_request=1**.)
/APPLIANCE_LIST_OUTPUT/REQUEST/RESOURCE	(#PCDATA)
	The resource specified for the request. (This element appears only when the API request includes the parameter **echo_request=1**.)
/APPLIANCE_LIST_OUTPUT/REQUEST/PARAM_LIST	(PARAM+)
/APPLIANCE_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM	(KEY, VALUE)
/APPLIANCE_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM/KEY	(#PCDATA)
	An input parameter name. (This element appears only when the API request includes the parameter **echo_request=1**.)
/APPLIANCE_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM/VALUE	(#PCDATA)
	An input parameter value. This element appears only when the API request includes the parameter **echo_request=1**.
/APPLIANCE_LIST_OUTPUT/REQUEST/POST_DATA	(#PCDATA)
	The POST data, if any. (This element appears only when the API request includes the parameter **echo_request=1**.)
/APPLIANCE_LIST_OUTPUT/RESPONSE	(DATETIME, (APPLIANCE_LIST?, LICENSE_INFO?)

XPath	element specifications / notes
/APPLIANCE_LIST_OUTPUT/RESPONSE/DATETIME (#PCDATA)	
	The date and time of the Qualys response.
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST (APPLIANCE+)	
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE	
	(ID, NAME, SOFTWARE_VERSION, RUNNING_SCAN_COUNT, STATUS, MODEL_NUMBER?, SERIAL_NUMBER?, ACTIVATION_CODE?, INTERFACE_SETTINGS*, PROXY_SETTINGS?, VLANS?, STATIC_ROUTES?, ML_LATEST?, ML_VERSION?, VULNSIGS_LATEST?, VULNSIGS_VERSION?, ASSET_GROUP_COUNT?, ASSET_GROUP_LIST?, LAST_UPDATED_DATE?, POLLING_INTERVAL?, USER_LOGIN?, HEARTBEATS_MISSED?, SS_CONNECTION?, SS_LAST_CONNECTED?, FDCC_ENABLED?, USER_LIST?, UPDATED?, COMMENTS?, RUNNING_SCANS?, MAX_CAPACITY_UNITS?)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/ID (#PCDATA)	
	The scanner appliance ID.
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/NAME (#PCDATA)	
	The friendly name of the scanner appliance.
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/SOFTWARE_VERSION (#PCDATA)	
	The scanner appliance system software, which is installed on the appliance itself.
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/RUNNING_SCAN_COUNT	
	(#PCDATA)
	The number of scans currently running on the scanner appliance.
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/STATUS (#PCDATA)	
	The scanner appliance heartbeat check status. “Online” indicates the appliance did not miss the most recent heartbeat check. “Offline” indicates the appliance missed one or more heartbeat checks because it did not contact the Security Operations Center. (Heartbeat checks occur every 4 hours.)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/MODEL_NUMBER (#PCDATA)	
	The model number of the scanner appliance. (This element appears only when the API request includes the parameter **`output_mode=full`**.)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/SERIAL_NUMBER (#PCDATA)	
	The serial number (ID) of the scanner appliance. (This element appears only when the API request includes the parameter **`output_mode=full`**.)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/ACTIVATION_CODE (#PCDATA)	
	The activation code provisioned for the scanner appliance. (This element appears only when the API request includes the parameter **`output_mode=full`**.)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/INTERFACE_SETTINGS	
	(SETTING?, INTERFACE, IP_ADDRESS, NETMASK, GATEWAY, LEASE, IPV6_ADDRESS?, SPEED, DUPLEX, DNS)

XPath	element specifications / notes
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/INTERFACE_SETTINGS/SETTING (#PCDATA)	
	A flag indicating whether the WAN interface is disabled. When the WAN interface is disabled, the value Disabled appears. When enabled, this element is not displayed . (This element appears only when the API request includes the parameter **output_mode=full**.)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/INTERFACE_SETTINGS/INTERFACE (#PCDATA)	
	The network interface: “lan” or “wan”. (This element appears only when the API request includes the parameter **output_mode=full**.)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/INTERFACE_SETTINGS/IP_ADDRESS (#PCDATA)	
	The LAN or WAN IP address. (This element appears only when the API request includes the parameter **output_mode=full**.)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/INTERFACE_SETTINGS/NETMASK (#PCDATA)	
	The netmask value for the LAN or WAN interface.. (This element appears only when the API request includes the parameter **output_mode=full**.)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/INTERFACE_SETTINGS/GATEWAY (#PCDATA)	
	The gateway IP address for the LAN or WAN interface. (This element appears only when the API request includes the parameter **output_mode=full**.)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/INTERFACE_SETTINGS/LEASE (#PCDATA)	
	The lease for the LAN or WAN interface: Static for a static IP address or Dynamic for DHCP. (This element appears only when the API request includes the parameter **output_mode=full**.)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/INTERFACE_SETTINGS/IPV6_ADDRESS (#PCDATA)	
	The LAN Pv6 address, if any. (This element appears only when the API request includes the parameter **output_mode=full**.)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/INTERFACE_SETTINGS/SPEED (#PCDATA)	
	The speed of the LAN or WAN interface. (This element appears only when the API request includes the parameter **output_mode=full**.)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/INTERFACE_SETTINGS/DUPLEX (#PCDATA)	
	The duplex setting for the LAN or WAN port links: Full Duplex, Half Duplex, or Unknown. (This element appears only when the API request includes the parameter **output_mode=full**.)

XPath	element specifications / notes
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/INTERFACE_SETTINGS/DNS (DOMAIN?, PRIMARY, SECONDARY)	
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/INTERFACE_SETTINGS/DNS/DOMAIN (#PCDATA)	
	The domain name of the DNS server. (This element appears only when the API request includes the parameter **output_mode=full**.)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/INTERFACE_SETTINGS/DNS/PRIMARY (#PCDATA)	
	The IP address of the primary DNS server. (This element appears only when the API request includes the parameter **output_mode=full**.)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/INTERFACE_SETTINGS/DNS/SECONDARY (#PCDATA)	
	The IP address of the secondary DNS server. (This element appears only when the API request includes the parameter **output_mode=full**.)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/PROXY_SETTINGS	
	(SETTING, PROXY*)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/PROXY_SETTINGS/PROXY	
	(IP_ADDRESS, PORT, USER)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/PROXY_SETTINGS/PROXY/IP_ADDRESS (#PCDATA)	
	The IP address of the primary proxy server. If this setting is disabled, Disabled appears. (This element appears only when the API request includes the parameter **output_mode=full**.)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/PROXY_SETTINGS/PROXY/PORT (#PCDATA)	
	The port number assigned to the primary proxy server. If this setting is disabled, Disabled appears. (This element appears only when the API request includes the parameter **output_mode=full**.)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/PROXY_SETTINGS/PROXY/USER (#PCDATA)	
	The user name for proxy authentication on the primary proxy server, if authentication is enabled. If this setting is disabled, Disabled appears. (This element appears only when the API request includes the parameter **output_mode=full**.)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/VLANS (SETTING, VLAN*)	
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/VLANS/SETTING (#PCDATA)	
	A flag indicating whether VLANS are enabled: “enabled” or “disabled”. (This element appears only when the API request includes the parameter **output_mode=full**.)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/VLANS/VLAN	
	(ID, NAME, IP_ADDRESS, NETMASK)

XPath	element specifications / notes
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/VLANS/VLAN/ ID (#PCDATA)	
	A VLAN ID. (This element appears only when the API request includes the parameter **output_mode=full**.)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/VLANS/VLAN/ NAME (#PCDATA)	
	A VLAN name. (This element appears only when the API request includes the parameter **output_mode=full**.)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/VLANS/VLAN/ IP_ADDRESS (#PCDATA)	
	A valid IP address for a VLAN. (This element appears only when the API request includes the parameter **output_mode=full**.)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/VLANS/VLAN/ NETMASK (#PCDATA)	
	A valid netmask for a VLAN. (This element appears only when the API request includes the parameter **output_mode=full**.)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/STATIC_ROUTES (ROUTE*)	
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/STATIC_ROUTES/ROUTE	
	(NAME, IP_ADDRESS, NETMASK, GATEWAY)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/STATIC_ROUTES/ROUTE/ NAME (#PCDATA)	
	A static route name. (This element appears only when the API request includes the parameter **output_mode=full**.)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/STATIC_ROUTES/ROUTE/ IP_ADRESS (#PCDATA)	
	A target network for a static route. (This element appears only when the API request includes the parameter **output_mode=full**.)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/STATIC_ROUTES/ROUTE/ NETMASK (#PCDATA)	
	A netmask for a static route. (This element appears only when the API request includes the parameter **output_mode=full**.)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/STATIC_ROUTES/ROUTE/ GATEWAY (#PCDATA)	
	A gateway IP address for a static route. (This element appears only when the API request includes the parameter **output_mode=full**.)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/ML_LATEST (#PCDATA)	
	The latest scanning engine version available. (This element appears only when the API request includes the parameter **output_mode=full**.)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/ML_VERSION (#PCDATA)	
	The scanning engine version currently installed on the scanner appliance. (This element appears only when the API request includes the parameter **output_mode=full**.)
attribute: updated	"yes" indicates the appliance is updated with the latest version.

XPath	element specifications / notes
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/VULNSIGS_LATEST (#PCDATA)	
	The latest vulnerability signatures version available. (This element appears only when the API request includes the parameter **output_mode=full**.)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/VULNSIGS_VERSION (#PCDATA)	
	The vulnerability signatures version currently installed on the scanner appliance. (This element appears only when the API request includes the parameter **output_mode=full**.)
attribute: updated	"yes" indicates the appliance is updated with the latest version.
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/ASSET_GROUP_COUNT (#PCDATA)	
	The number of asset groups that the scanner appliance belongs to. (This element appears only when the API request includes the parameter **output_mode=full**.)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/ASSET_GROUP_LIST (ASSET_GROUP*)	
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/ASSET_GROUP_LIST/ ASSET_GROUP (ID, NAME)	
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/ASSET_GROUP_LIST/ ASSET_GROUP/ID (#PCDATA)	
	The ID of an asset group that the appliance belongs to. (This element appears only when the API request includes the parameter **output_mode=full**.)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/ASSET_GROUP_LIST/ ASSET_GROUP/NAME (#PCDATA)	
	The name of an asset group that the appliance belongs to. (This element appears only when the API request includes the parameter **output_mode=full**.)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/LAST_UPDATED_DATE (#PCDATA)	
	The last date and time when the scanner appliance received a software update. (This element appears only when the API request includes the parameter **output_mode=full**.)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/POLLING_INTERVAL (#PCDATA)	
	The polling interval defined for the scanner appliance. (This element appears only when the API request includes the parameter **output_mode=full**.)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/USER_LOGIN (#PCDATA)	
	The user login. (This element appears only when the API request includes the parameter **output_mode=full**.)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/HEARTBEATS_MISSED (#PCDATA)	
	The number of heartbeat checks missed. (This element appears only when the API request includes the parameter **output_mode=full**.)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/SS_CONNECTION (#PCDATA)	
	The new scanner services status: connected or not connected. (This element appears only when the API request includes the parameter **output_mode=full**.)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/SS_LAST_CONNECTED (#PCDATA)	
	The last date/time when new scanner services connected. (This element appears only when the API request includes the parameter **output_mode=full**.)

XPath	element specifications / notes
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/FDCC_ENABLED (#PCDATA)	
	A flag indicating whether the FDCC module is enabled on the appliance.
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/UPDATED (#PCDATA)	
	A flag indicating whether the appliance is updated with the latest scanning engine software and vulnerability signatures software: "yes" or "no". (This element appears only when the API request includes the parameter **output_mode=full**.)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/RUNNING_SCANS (SCAN+)	
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/RUNNING_SCANS/SCAN	
	(ID, TITLE, REF, TYPE, SCAN_DATE)
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/RUNNING_SCANS/SCAN/ ID (#PCDATA)	
	The scan ID of a currently scan running on the scanner appliance.
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/RUNNING_SCANS/SCAN/ TITLE (#PCDATA)	
	The title of a currently scan running on the scanner appliance.
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/RUNNING_SCANS/SCAN/ REF (#PCDATA)	
	The scan reference ID for a currently scan running on the scanner appliance.
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/RUNNING_SCANS/SCAN/ TYPE (#PCDATA)	
	The scan type of a scan currently running on the scanner appliance. The scan type will be one of: Vulnerability Scan, Compliance Scan, Web Application Scan, FDCC Scan, or Map.
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/RUNNING_SCANS/SCAN/ SCAN_DATE (#PCDATA)	
	The date and time when the currently running scan was launched.
/APPLIANCE_LIST_OUTPUT/RESPONSE/APPLIANCE_LIST/APPLIANCE/MAX_CAPACITY_UNITS (#PCDATA)	
	The percentage of capacity available for the scanner appliance. (This element appears only when the API request includes the parameter **output_mode=full**.)
/APPLIANCE_LIST_OUTPUT/RESPONSE/LICENSE_INFO	
	(QVSA_LICENSES_COUNT, QVSA_LICENSES_USED)
/APPLIANCE_LIST_OUTPUT/RESPONSE/LICENSE_INFO /QVSA_LICENSES_COUNT (#PCDATA)	
	The number of virtual scanner licenses available in your account.
/APPLIANCE_LIST_OUTPUT/RESPONSE/LICENSE_INFO /QVSA_LICENSES_USED (#PCDATA)	
	The number of virtual scanner licenses that have been used.

Scanner Appliance Create Output

The Scanner Appliance List API v2 (/api/2.0/fo/appliance/ with action=create) returns XML output using the "appliance _create_output.dtd". This DTD can be found at the following URL (where qualysapi.qualys.com is the base URL of the user's Qualys platform):

```
https://qualysapi.qualys.com/api/2.0/fo/appliance/
appliance_create_output.dtd
```

DTD for Scanner Appliance Create Output

A recent DTD for the scanner appliance create output (appliance_create_output.dtd) is shown below.

```
<!-- QUALYS APPLIANCE_CREATE_OUTPUT DTD -->
<!ELEMENT APPLIANCE_CREATE_OUTPUT (REQUEST?,RESPONSE)>

<!ELEMENT REQUEST (DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?,
                   POST_DATA?)>
<!ELEMENT DATETIME (#PCDATA)>
<!ELEMENT USER_LOGIN (#PCDATA)>
<!ELEMENT RESOURCE (#PCDATA)>
<!ELEMENT PARAM_LIST (PARAM+)>
<!ELEMENT PARAM (KEY, VALUE)>
<!ELEMENT KEY (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>
<!-- if returned, POST_DATA will be urlencoded -->
<!ELEMENT POST_DATA (#PCDATA)>

<!ELEMENT RESPONSE (DATETIME, APPLIANCE)>

<!ELEMENT APPLIANCE (ID, FRIENDLY_NAME, ACTIVATION_CODE,
                     REMAINING_QVSA_LICENSES)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT FRIENDLY_NAME (#PCDATA)>
<!ELEMENT ACTIVATION_CODE (#PCDATA)>
<!ELEMENT REMAINING_QVSA_LICENSES (#PCDATA)>
```

XPaths for Scanner Appliance Create Output

This section describes the XPaths for the scanner appliance create output.

XPath	element specifications / notes
/APPLIANCE_CREATE_OUTPUT	(REQUEST?,RESPONSE)
/APPLIANCE_CREATE_OUTPUT/REQUEST	(DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?, POST_DATA?)
/APPLIANCE_CREATE_OUTPUT/REQUEST/DATETIME	(#PCDATA) The date and time of the API request. (This element appears only when the API request includes the parameter **echo_request=1**.)
/APPLIANCE_CREATE_OUTPUT/REQUEST/USER_LOGIN	(#PCDATA) The user login ID of the user who made the request. (This element appears only when the API request includes the parameter **echo_request=1**.)
/APPLIANCE_CREATE_OUTPUT/REQUEST/RESOURCE	(#PCDATA) The resource specified for the request. (This element appears only when the API request includes the parameter **echo_request=1**.)
/APPLIANCE_CREATE_OUTPUT/REQUEST/PARAM_LIST	(PARAM+)
/APPLIANCE_CREATE_OUTPUT/REQUEST/PARAM_LIST/PARAM	(KEY, VALUE)
/APPLIANCE_CREATE_OUTPUT/REQUEST/PARAM_LIST/PARAM/KEY	(#PCDATA) An input parameter name. (This element appears only when the API request includes the parameter **echo_request=1**.)
/APPLIANCE_CREATE_OUTPUT/REQUEST/PARAM_LIST/PARAM/VALUE	(#PCDATA) An input parameter value. This element appears only when the API request includes the parameter **echo_request=1**.
/APPLIANCE_CREATE_OUTPUT/REQUEST/POST_DATA	(#PCDATA) The POST data, if any. (This element appears only when the API request includes the parameter **echo_request=1**.)
/APPLIANCE_CREATE_OUTPUT/RESPONSE	(DATETIME, APPLIANCE)
/APPLIANCE_CREATE_OUTPUT/RESPONSE/DATETIME	(#PCDATA) The date and time of the Qualys response.
/APPLIANCE_CREATE_OUTPUT/RESPONSE/APPLIANCE	(ID, FRIENDLY_NAME, ACTIVATION_CODE, REMAINING_QVSA_LICENSES)
/APPLIANCE_CREATE_OUTPUT/RESPONSE/APPLIANCE/ID	(#PCDATA) The scanner appliance ID.
/APPLIANCE_CREATE_OUTPUT/RESPONSE/APPLIANCE/FRIENDLY_NAME	(#PCDATA) The friendly name of the scanner appliance.
/APPLIANCE_CREATE_OUTPUT/RESPONSE/APPLIANCE/ACTIVATION_CODE	(#PCDATA) The activation code for the scanner appliance.
/APPLIANCE_CREATE_OUTPUT/RESPONSE/APPLIANCE/REMAINING_QVSA_LICENSES	(#PCDATA) The number of remaining virtual scanner license in your account.

KnowledgeBase Output

The KnowledgeBase XML output provides information about vulnerabilities in the Qualys KnowledgeBase. The KnolwedgeBase output is returned from a KnowledgeBase API request.

The DTD "knowledge_base_vuln_list_output.dtd" can be found at the following URL:

`https://qualysapi.qualys.com/api/2.0/fo/knowledge_base/vuln/knowledge_base_vuln_list_output.dtd`

where <qualysapi.qualys.com> is the API server URL where your account is located.

DTD for KnowledgeBase Output

A recent DTD for the KnowledgeBase output is shown below.

```
<!-- QUALYS KNOWLEDGE_BASE_VULN_LIST_OUTPUT DTD -->
<!ELEMENT KNOWLEDGE_BASE_VULN_LIST_OUTPUT (REQUEST?,RESPONSE)>

  <!ELEMENT REQUEST (DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?,
                     POST_DATA?)>
    <!ELEMENT DATETIME (#PCDATA)>
    <!ELEMENT USER_LOGIN (#PCDATA)>
    <!ELEMENT RESOURCE (#PCDATA)>
    <!ELEMENT PARAM_LIST (PARAM+)>
      <!ELEMENT PARAM (KEY, VALUE)>
        <!ELEMENT KEY (#PCDATA)>
        <!ELEMENT VALUE (#PCDATA)>
    <!-- if returned, POST_DATA will be urlencoded -->
    <!ELEMENT POST_DATA (#PCDATA)>

  <!ELEMENT RESPONSE (DATETIME, (VULN_LIST|ID_SET)?, WARNING?)>
    <!-- DATETIME already defined -->
    <!ELEMENT VULN_LIST (VULN+)>
      <!ELEMENT VULN (QID, VULN_TYPE, SEVERITY_LEVEL, TITLE, CATEGORY,
            LAST_CUSTOMIZATION?, LAST_SERVICE_MODIFICATION_DATETIME?,
            PUBLISHED_DATETIME, BUGTRAQ_LIST?, PATCHABLE, SOFTWARE_LIST?,
            VENDOR_REFERENCE_LIST?, CVE_LIST?,
            DIAGNOSIS?, DIAGNOSIS_COMMENT?, CONSEQUENCE?,
            CONSEQUENCE_COMMENT?, SOLUTION?, SOLUTION_COMMENT?,
            COMPLIANCE_LIST?, CORRELATION?, CVSS?, PCI_FLAG, PCI_REASONS?,
            DISCOVERY)>

        <!ELEMENT QID (#PCDATA)>
        <!ELEMENT VULN_TYPE (#PCDATA)>
        <!ELEMENT SEVERITY_LEVEL (#PCDATA)>
```

```
          <!ELEMENT TITLE (#PCDATA)>
          <!ELEMENT CATEGORY (#PCDATA)>
          <!ELEMENT LAST_CUSTOMIZATION (DATETIME, USER_LOGIN?)>
            <!-- USER_LOGIN already defined (no USER_LOGIN for OVAL Vulns) -
->
          <!ELEMENT LAST_SERVICE_MODIFICATION_DATETIME (#PCDATA)>
          <!ELEMENT PUBLISHED_DATETIME (#PCDATA)>
          <!ELEMENT BUGTRAQ_LIST (BUGTRAQ+)>
            <!ELEMENT BUGTRAQ (ID, URL)>
              <!ELEMENT ID (#PCDATA)>
              <!ELEMENT URL (#PCDATA)>
          <!ELEMENT PATCHABLE (#PCDATA)>
          <!ELEMENT SOFTWARE_LIST (SOFTWARE+)>
            <!ELEMENT SOFTWARE (PRODUCT, VENDOR)>
              <!ELEMENT PRODUCT (#PCDATA)>
              <!ELEMENT VENDOR (#PCDATA)>
          <!ELEMENT VENDOR_REFERENCE_LIST (VENDOR_REFERENCE+)>
            <!ELEMENT VENDOR_REFERENCE (ID, URL)>
          <!ELEMENT CVE_LIST (CVE+)>
            <!ELEMENT CVE (ID, URL)>
            <!-- ID, URL already defined -->
          <!ELEMENT DIAGNOSIS (#PCDATA)>
          <!ELEMENT DIAGNOSIS_COMMENT (#PCDATA)>
          <!ELEMENT CONSEQUENCE (#PCDATA)>
          <!ELEMENT CONSEQUENCE_COMMENT (#PCDATA)>
          <!ELEMENT SOLUTION (#PCDATA)>
          <!ELEMENT SOLUTION_COMMENT (#PCDATA)>
          <!ELEMENT COMPLIANCE_LIST (COMPLIANCE+)>
            <!ELEMENT COMPLIANCE (TYPE, SECTION, DESCRIPTION)>
              <!ELEMENT TYPE (#PCDATA)>
              <!ELEMENT SECTION (#PCDATA)>
              <!ELEMENT DESCRIPTION (#PCDATA)>
          <!ELEMENT CORRELATION (EXPLOITS?, MALWARE?)>
            <!ELEMENT EXPLOITS (EXPLT_SRC+)>
              <!ELEMENT EXPLT_SRC (SRC_NAME, EXPLT_LIST)>
                <!ELEMENT SRC_NAME (#PCDATA)>
                <!ELEMENT EXPLT_LIST (EXPLT+)>
                  <!ELEMENT EXPLT (REF, DESC, LINK?)>
                    <!ELEMENT REF (#PCDATA)>
                    <!ELEMENT DESC (#PCDATA)>
                    <!ELEMENT LINK (#PCDATA)>
            <!ELEMENT MALWARE (MW_SRC+)>
              <!ELEMENT MW_SRC (SRC_NAME, MW_LIST)>
                <!ELEMENT MW_LIST (MW_INFO+)>
                  <!ELEMENT MW_INFO (MW_ID, MW_TYPE?, MW_PLATFORM?,
                                     MW_ALIAS?, MW_RATING?, MW_LINK?)>
                    <!ELEMENT MW_ID (#PCDATA)>
                    <!ELEMENT MW_TYPE (#PCDATA)>
                    <!ELEMENT MW_PLATFORM (#PCDATA)>
```

```
                    <!ELEMENT MW_ALIAS (#PCDATA)>
                    <!ELEMENT MW_RATING (#PCDATA)>
                    <!ELEMENT MW_LINK (#PCDATA)>
          <!ELEMENT CVSS (BASE, TEMPORAL?, ACCESS?, IMPACT?,
                          AUTHENTICATION?, EXPLOITABILITY?,
                          REMEDIATION_LEVEL?, REPORT_CONFIDENCE?)>
            <!ELEMENT BASE (#PCDATA)>
              <!ATTLIST BASE source CDATA #IMPLIED>
            <!ELEMENT TEMPORAL (#PCDATA)>
            <!ELEMENT ACCESS (VECTOR?, COMPLEXITY?)>
              <!ELEMENT VECTOR (#PCDATA)>
              <!ELEMENT COMPLEXITY (#PCDATA)>
            <!ELEMENT IMPACT (CONFIDENTIALITY?, INTEGRITY?, AVAILABILITY?)>
              <!ELEMENT CONFIDENTIALITY (#PCDATA)>
              <!ELEMENT INTEGRITY (#PCDATA)>
              <!ELEMENT AVAILABILITY (#PCDATA)>
            <!ELEMENT AUTHENTICATION (#PCDATA)>
            <!ELEMENT EXPLOITABILITY (#PCDATA)>
            <!ELEMENT REMEDIATION_LEVEL (#PCDATA)>
            <!ELEMENT REPORT_CONFIDENCE (#PCDATA)>

          <!ELEMENT PCI_FLAG (#PCDATA)>
          <!ELEMENT PCI_REASONS (PCI_REASON+)>
          <!ELEMENT PCI_REASON (#PCDATA)>

          <!ELEMENT DISCOVERY (REMOTE, AUTH_TYPE_LIST?)>
            <!ELEMENT REMOTE (#PCDATA)>
            <!ELEMENT AUTH_TYPE_LIST (AUTH_TYPE+)>
              <!ELEMENT AUTH_TYPE (#PCDATA)>

      <!ELEMENT ID_SET ((ID|ID_RANGE)+)>
        <!-- ID already defined -->
        <!ELEMENT ID_RANGE (#PCDATA)>

      <!ELEMENT WARNING (CODE?, TEXT, URL?)>
        <!ELEMENT CODE (#PCDATA)>
        <!ELEMENT TEXT (#PCDATA)>
        <!-- URL already defined -->

<!-- EOF -->
```

XPaths for KnowledgeBase Output

This section describes the XPaths for the KnowledgeBase output.

XPath	element specifications / notes
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT	(REQUEST?, RESPONSE)
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/REQUEST	(DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?, POST_DATA?)
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/REQUEST/DATETIME	(#PCDATA)
	The date and time of the API request. (This element appears only when the API request includes the parameter **echo_request=1**.)
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/REQUEST/USER_LOGIN	(#PCDATA)
	The user login ID of the user who made the request. (This element appears only when the API request includes the parameter **echo_request=1**.)
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/REQUEST/RESOURCE	(#PCDATA)
	The resource specified for the request. (This element appears only when the API request includes the parameter **echo_request=1**.)
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/REQUEST/PARAM_LIST	(PARAM+))
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM	(KEY, VALUE))
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM/KEY	(#PCDATA)
	An input parameter name. (This element appears only when the API request includes the parameter **echo_request=1**.)
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM/VALUE	(#PCDATA)
	An input parameter value. This element appears only when the API request includes the parameter **echo_request=1**.
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/REQUEST/POST_DATA	(#PCDATA)
	The POST data, if any. (This element appears only when the API request includes the parameter **echo_request=1**.)
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE	(DATETIME, (VULN_LIST|ID_SET)?, WARNING?)
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/DATETIME	(#PCDATA)
	The date and time of the Qualys response.
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST	(VULN+)
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN	(QID, VULN_TYPE, SEVERITY_LEVEL, TITLE, CATEGORY, LAST_CUSTOMIZATION?, LAST_SERVICE_MODIFICATION_DATETIME?, PUBLISHED_DATETIME, BUGTRAQ_LIST?, PATCHABLE, SOFTWARE_LIST?, VENDOR_REFERENCE_LIST?, CVE_LIST?, DIAGNOSIS?, DIAGNOSIS_COMMENT?, CONSEQUENCE?, CONSEQUENCE_COMMENT?, SOLUTION?, SOLUTION_COMMENT?, COMPLIANCE_LIST?, CORRELATION?, CVSS?, PCI_FLAG, PCI_REASONS?, DISCOVERY)
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/QID	(#PCDATA)
	The vulnerability QID (Qualys ID), assigned by the service.

XPath	element specifications / notes
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/VULN_TYPE (#PCDATA)	
	The vulnerability type: Vulnerability, Potential Vulnerability or Information Gathered. The type "Vulnerability or Potential Vulnerability" corresponds to the half red/half yellow icon in the QualyGuard user interface. If confirmed to exist on a host during a scan, the vulnerability is classified as a confirmed vulnerability in your account; if not the vulnerability is classified as a potential vulnerability in your account.
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/SEVERITY_LEVEL (#PCDATA)	
	The severity level of the vulnerability. A valid value for a confirmed or potential vulnerability is an integer 1 to 5, where 5 represents the most serious risk if exploited. A valid value for information gathered is a value 1 to 3, where 3 represents the most serious risk if exploited.
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/TITLE (#PCDATA)	
	The vulnerability title.
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/CATEGORY (#PCDATA)	
	The vulnerability category.
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/ LAST_CUSTOMIZATION (DATETIME, USER_LOGIN)	
	The date this vulnerability was last customized by a user, in YYYY-MM-DDTHH:MM:SSZ format (UTC/GMT).
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/ LAST_SERVICE_MODIFIDATION_DATETIME (#PCDATA)	
	The date this vulnerability was last updated by the service, in YYYY-MM-DDTHH:MM:SSZ format (UTC/GMT).
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/ PUBLISHED_DATETIME (#PCDATA)	
	The date this vulnerability was published by the service, in YYYY-MM-DDTHH:MM:SSZ format (UTC/GMT).
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/ BUGTRAQ_LIST (BUGTRAQ+)	
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/ BUGTRAQ_LIST/BUGTRAQ (ID, URL)	
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/ BUGTRAQ_LIST/BUGTRAQ/ID (#PCDATA)	
	A Bugtraq ID for a vulnerability.
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/ BUGTRAQ_LIST/BUGTRAQ/URL (#PCDATA)	
	The URL to a Bugtraq ID.
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/ PATCHABLE (#PCDATA)	
	A flag indicating whether there is a patch available to fix the vulnerability. The value 1 indicates a patch is available to fix the vulnerability. The value 0 indicates a patch is not available to fix the vulnerability.

XPath	element specifications / notes
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/SOFTWARE_LIST (SOFTWARE+)	
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/SOFTWARE_LIST/SOFTWARE (PRODUCT, VENDOR)	
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/SOFTWARE_LIST/SOFTWARE/PRODUCT (#PCDATA)	
	Software product information associated with the vulnerability. This information is provided by NIST as a part of CVE information. (This element appears only when the API request includes the parameter **`details=All`**.)
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/SOFTWARE_LIST/SOFTWARE/VENDOR (#PCDATA)	
	Software vendor information associated with the vulnerability. This information is provided by NIST as a part of CVE information. (This element appears only when the API request includes the parameter **`details=All`**.)
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/VENDOR_REFERENCE_LIST (VENDOR, REFERENCE+)	
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/VENDOR_REFERENCE_LIST/VENDOR (ID, URL)	
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/VENDOR_REFERENCE_LIST/VENDOR/ID (#PCDATA)	
	A name of a vendor reference.
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/VENDOR_REFERENCE_LIST/VENDOR/URL (#PCDATA)	
	The URL to a vendor reference.
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/CVE (ID, URL)	
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/CVE/ID (#PCDATA)	
	A CVE name assigned to the vulnerability. CVE (Common Vulnerabilities and Exposures) is a list of common names for publicly known vulnerabilities and exposures. Through open and collaborative discussions, the CVE Editorial Board determines which vulnerabilities or exposures are included in CVE. If the CVE name starts with CAN (candidate) then it is under consideration for entry into CVE.
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/CVE/URL (#PCDATA)	
	The URL to a CVE name.
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/DIAGNOSIS (#PCDATA)	
	A service-provided description of the threat posed by the vulnerability if successfully exploited.

XPath	element specifications / notes
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/DIAGNOSIS_COMMENT (#PCDATA)	
	A user-customized description of the threat posed by the vulnerability if successfully exploited.
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/CONSEQUENCE (#PCDATA)	
	A service-provided description of the consequences that may occur if this vulnerability is successfully exploited.
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/CONSEQUENCE_COMMENT (#PCDATA)	
	A user-customized description of the consequences that may occur if this vulnerability is successfully exploited.
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/SOLUTION (#PCDATA)	
	A service-provided description of a verified solution to fix the vulnerability.
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/SOLUTION_COMMENT (#PCDATA)	
	A user-customized description of a verified solution to fix the vulnerability.
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/COMPLIANCE_LIST (COMPLIANCE+)	
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/COMPLIANCE_LIST (TYPE, SECTION, DESCRIPTION)	
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/COMPLIANCE_LIST/TYPE (#PCDATA)	
	A type of a compliance information associated with the vulnerability: HIPAA, GLBA, CobIT or SOX.
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/COMPLIANCE_LIST/SECTION (#PCDATA)	
	A section of a compliance policy or regulation.
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/COMPLIANCE_LIST/DESCRIPTION (#PCDATA)	
	A description of a compliance policy or regulation.
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/CORRELATION (EXPLOITS?, MALWARE?)	
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/CORRELATION/EXPLOITS (EXPL_SRC+)	
	The <EXPLOITS> element and its sub-elements appear only when there is exploitability information for the vulnerability from third party vendors and/or publicly available sources.

XPath	element specifications / notes
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/CORRELATION/EXPLOITS/EXPL_SRC (SRC_NAME, EXPLT_LIST)	
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/CORRELATION/EXPLOITS/EXPL_SRC/SRC_NAME (#PCDATA)	
	A name of a third party vendor or publicly available source whose exploitability information is correlated with the vulnerability.
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/CORRELATION/EXPLOITS/EXPL_SRC/EXPLT_LIST (EXPLT+)	
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/CORRELATION/EXPLOITS/EXPL_SRC/EXPLT_LIST/EXPLT (REF, DESC, LINK?)	
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/CORRELATION/EXPLOITS/EXPL_SRC/EXPLT_LIST/EXPLT/REF (#PCDATA)	
	A CVE reference for the exploitability information.
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/CORRELATION/EXPLOITS/EXPL_SRC/EXPLT_LIST/EXPLT/DESC (#PCDATA)	
	A description of the exploitability information provided by the source (third party vendor or publicly available source).
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/CORRELATION/EXPLOITS/EXPL_SRC/EXPLT_LIST/EXPLT/LINK (#PCDATA)	
	A link to the exploit for the vulnerability, when available from the source.
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/MALWARE (MW_SRC+)	
	The <MALWARE> element and its sub-elements appear only when there is malware information for the vulnerability from Trend Micro.
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/MALWARE/MW_SRC (SRC_NAME, MW_LIST)	
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/MALWARE/MW_SRC/SRC_NAME (#PCDATA)	
	The name of the source of the malware information: Trend Micro.
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/MALWARE/MW_SRC/MW_LIST (MW_INFO+)	
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/MALWARE/MW_SRC/MW_LIST/MW_INFO (MW_ID, MW_TYPE?, MW_PLATFORM?, MW_ALIAS?, MW_RATING?, MW_LINK?)	
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/MALWARE/MW_SRC/MW_LIST/MW_INFO/MW_ID (#PCDATA)	
	A malware name/ID assigned by Trend Micro.
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/MALWARE/MW_SRC/MW_LIST/MW_INFO/MW_TYPE (#PCDATA)	
	A type of malware, such as Backdoor, Virus, Worm or Trojan.

XPath	element specifications / notes
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/MALWARE/MW_SRC/MW_LIST/MW_INFO/MW_PLATFORM (#PCDATA)	
	A list of the platforms that may be affected.
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/MALWARE/MW_SRC/MW_LIST/MW_INFO/MW_ALIAS (#PCDATA)	
	A list of other names used by different vendors and/or publicly available sources that refer to the same threat.
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/MALWARE/MW_SRC/MW_LIST/MW_INFO/MW_RATING (#PCDATA)	
	An overall risk rating as determined by Trend Micro: Low, Medium or High.
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/MALWARE/MW_SRC/MW_LIST/MW_INFO/MW_LINK (#PCDATA)	
	A link to malware details.
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/CVSS	(BASE, TEMPORAL?, ACCESS?, IMPACT?, AUTHENTICATION?, EXPLOITABILITY?, REMEDIATION_LEVEL?, REPORT_CONFIDENCE?)
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/CVSS/BASE (#PCDATA)	
	A CVSS base score assigned to the vulnerability. (This element appears only when the CVSS Scoring feature is turned on in the user's subscription and the API request is for "Basic" details or "All" details.)
attribute: **source**	**source** is *implied* and, if present, is "service" to indicate that the CVSS base score for the vulnerability is not supplied by NIST, as published in the National Vulnerability Database (NVD). The service displays a CVSS base score provided by NIST whenever available. In a case where NIST lists a CVSS base score of 0 or does not provide a score for a vulnerability in the NVD, the service determines whether the severity of the vulnerability warrants a higher CVSS base score. If so, a service generated score is provided and the attribute "source=service" appears in the XML output.
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/CVSS/TEMPORAL (#PCDATA)	
	A CVSS temporal score. (This element appears only when the CVSS Scoring feature is turned on in the user's subscription and the API request is for "Basic" details or "All" details.)
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/CVSS/ACCESS (VECTOR?, COMPLEXITY?)	
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/CVSS/ACCESS/VECTOR (#PCDATA)	
	A CVSS access vector metric. This metric reflects how the vulnerability is exploited. The more remote an attacker can be to attack a host, the greater the vulnerability score. The value is one of the following: Network, Adjacent Network, Local Access, or Undefined. See "CVSS V2 Sub Metrics Mapping" below. (This element appears only when the CVSS Scoring feature is turned on in the user's subscription and the API request includes the parameter **details=All**.)

XPath	element specifications / notes
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/CVSS/ACCESS/COMPLEXITY (#PCDATA)	A CVSS access complexity metric. This metric measures the complexity of the attack required to exploit the vulnerability once an attacker has gained access to the target system. The value is one of the following: Undefined, Low, Medium, or High. See "CVSS V2 Sub Metrics Mapping" below. (This element appears only when the CVSS Scoring feature is turned on in the user's subscription and the API request includes the parameter **`details=All`**.)
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/CVSS/IMPACT (CONFIDENTIALITY?, INTEGRITY?, AVAILABILITY?)	
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/CVSS/IMPACT/CONFIDENTIALITY (#PCDATA)	A CVSS confidentiality impact metric. This metric measures the impact on confidentiality of a successfully exploited vulnerability. The value is: Undefined, None, Partial, or Complete. See "CVSS V2 Sub Metrics Mapping" below. (This element appears only when the CVSS Scoring feature is turned on in the user's subscription and the API request includes the parameter **`details=All`**.)
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/CVSS/IMPACT/INTEGRITY (#PCDATA)	A CVSS integrity impact metric. This metric measures the impact to integrity of a successfully exploited vulnerability. The value is: Undefined, None, Partial, or Complete. See "CVSS V2 Sub Metrics Mapping" below. (This element appears only when the CVSS Scoring feature is turned on in the user's subscription and the API request includes the parameter **`details=All`**.)
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/CVSS/IMPACT/AVAILABILITY (#PCDATA)	A CVSS availability impact metric. This metric measures the impact to availability of a successfully exploited vulnerability. The value is: Undefined, None, Partial, or Complete. See "CVSS V2 Sub Metrics Mapping" below. (This element appears only when the CVSS Scoring feature is turned on in the user's subscription and the API request includes the parameter **`details=All`**.)
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/CVSS/AUTHENTICATION (#PCDATA)	A CVSS authentication metric. This metric measures the number of times an attacker must authenticate to a target in order to exploit a vulnerability. The value is: Undefined, Non required, Require single instance, or Require multiple instances. See "CVSS V2 Sub Metrics Mapping" below. (This element appears only when the CVSS Scoring feature is turned on in the user's subscription and the API request includes the parameter **`details=All`**.)
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/CVSS/EXPLOITABILITY (#PCDATA)	A CVSS exploitability metric. This metric measures the current state of exploit techniques or code availability. The value is: Undefined, Unproven, Proof-of-concept, Functional, or Widespread. See "CVSS V2 Sub Metrics Mapping" below. (This element appears only when the CVSS Scoring feature is turned on in the user's subscription and the API request includes the parameter **`details=All`**.)

XPath	element specifications / notes
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/CVSS/REMEDIATION_LEVEL (#PCDATA)	
	A CVSS remediation level metric. The remediation level of a vulnerability is an important factor for prioritization. The value is: Undefined, Official-fix, Temporary-fix, Workaround, or Unavailable. See "CVSS V2 Sub Metrics Mapping" below. (This element appears only when the CVSS Scoring feature is turned on in the user's subscription and the API request includes the parameter **`details=All`**.)
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/CVSS/REPORT_CONFIDENCE (#PCDATA)	
	A CVSS report confidence metric. This metric measures the degree of confidence in the existence of the vulnerability and the credibility of the known technical details. The value is: Undefined, Not confirmed, Uncorroborated, or Confirmed. See "CVSS V2 Sub Metrics Mapping" below. (This element appears only when the CVSS Scoring feature is turned on in the user's subscription and the API request includes the parameter **`details=All`**.)
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/PCI_FLAG (#PCDATA)	
	A flag indicating whether the vulnerability must be fixed to pass PCI compliance. The value 1 indicates the vulnerability must be fixed to pass PCI compliance. The value 0 indicates the vulnerability does not need to be fixed to pass PCI compliance.
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/PCI_REASONS (PCI_REASON+)	
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/PCI_REASONS/PCI_REASON (#PCDATA)	
	A reason why the vulnerability passed or failed PCI compliance. This appears only when the CVSS Scoring feature is turned on in the user's subscription and the API request includes the parameter **`show_pci_reasons=1`**.
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/DISCOVERY (REMOTE, AUTH_TYPE_LIST?))	
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/DISCOVERY/REMOTE (#PCDATA)	
	A flag indicating whether the discovery method is remotely detectable. The value 0 indicates the vulnerability cannot be detected remotely (authentication is required). The value 1 indicates the vulnerability can be detected in two ways: 1) remotely without using authentication, and 2) using authentication.
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/DISCOVERY/AUTH_TYPE_LIST (AUTH_TYPE+)	
/KNOWLEDGE_BASE_VULN_LIST_OUTPUT/RESPONSE/VULN_LIST/VULN/DISCOVERY/AUTH_TYPE_LIST/AUTH_TYPE (#PCDATA)	
	An authentication type used to detect the vulnerability using trusted scanning.

CVSS V2 Sub Metrics Mapping

A mapping of the CVSS sub metric values, as returned in the KnowledgeBase output, and the CVSS sub metric names, as defined by the CVSS standard, is provided below.

Base Family

Metric Value	KnowledgeBase Output XML Element and Value
Access Vector (AV)	
Local (L)	<VECTOR>1</VECTOR>
Adjacent Network (A)	<VECTOR>2</VECTOR>
Network (N)	<VECTOR>3</VECTOR>
Access Complexity	
Low (L)	<COMPLEXITY>1</COMPLEXITY>
Medium (M)	<COMPLEXITY>2</COMPLEXITY>
High (H)	<COMPLEXITY>3</COMPLEXITY>
Authentication (Au)	
None (N)	<AUTHENTICATION>1</AUTHENTICATION>
Single (S)	<AUTHENTICATION>2</AUTHENTICATION>
Multiple (M)	<AUTHENTICATION>3</AUTHENTICATION>
Confidentiality Impact (C)	
None (N)	<CONFIDENTIALITY>1</CONFIDENTIALITY>
Partial (P)	<CONFIDENTIALITY>2</CONFIDENTIALITY>
Complete (C)	<CONFIDENTIALITY>3</CONFIDENTIALITY>
Integrity Impact (I)	
None (N)	<INTEGRITY>1</INTEGRITY>
Partial (P)	<INTEGRITY>2</INTEGRITY>
Complete (C)	<INTEGRITY>3</INTEGRITY>
Availability Impact (A)	
None (N)	<AVAILABILITY>1</AVAILABILITY>
Partial (P)	<AVAILABILITY>2</AVAILABILITY>
Complete (C)	<AVAILABILITY>3</AVAILABILITY>

Temporal Metrics Family

Metric Value	KnowledgeBase Download XML Element and Value
Exploitability (E)	
Unproven (U)	<EXPLOITABILITY>1</EXPLOITABILITY>
Proof-of-Concept (POC)	<EXPLOITABILITY>2</EXPLOITABILITY>
Functional (F)	<EXPLOITABILITY>3</EXPLOITABILITY>
High (H)	<EXPLOITABILITY>4</EXPLOITABILITY>
Not Defined (ND)	<EXPLOITABILITY>0</EXPLOITABILITY>
Remediation Level (RL)	
Official Fix (OF)	<REMEDIATION_LEVEL>1</REMEDIATION_LEVEL>
Temporary Fix (TF)	<REMEDIATION_LEVEL>2</REMEDIATION_LEVEL>
Workaround (W)	<REMEDIATION_LEVEL>3</REMEDIATION_LEVEL>
Unavailable (U)	<REMEDIATION_LEVEL>4</REMEDIATION_LEVEL>
Not Defined (ND)	<REMEDIATION_LEVEL>0</REMEDIATION_LEVEL>
Report Confidence (RC)	
Unconfirmed (UC)	<REPORT_CONFIDENCE>1</REPORT_CONFIDENCE>
Uncorroborated (UR)	<REPORT_CONFIDENCE>2</REPORT_CONFIDENCE>
Confirmed (C)	<REPORT_CONFIDENCE>3</REPORT_CONFIDENCE>
Not Defined (ND)	<REPORT_CONFIDENCE>0</REPORT_CONFIDENCE>

Report List

The report list output is an XML report returned from the report list API call.

The DTD "report_list_output.dtd" can be found at the following URL (where <qualysapi.qualys.com> is the API server URL where your account is located):

```
https://qualysapi.qualys.com/api/2.0/fo/report/
report_list_output.dtd
```

DTD for Report List Output

A recent DTD for the report list output (report_list_output.dtd) is shown below.

```
<!-- QUALYS REPORT_LIST_OUTPUT DTD -->
<!ELEMENT REPORT_LIST_OUTPUT (REQUEST?, RESPONSE)>
<!ELEMENT REQUEST (DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?,
                   POST_DATA?)>
<!ELEMENT DATETIME (#PCDATA)>
<!ELEMENT USER_LOGIN (#PCDATA)>
<!ELEMENT RESOURCE (#PCDATA)>
<!ELEMENT PARAM_LIST (PARAM+)>
<!ELEMENT PARAM (KEY, VALUE)>
<!ELEMENT KEY (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>
<!-- if returned, POST_DATA will be urlencoded -->
<!ELEMENT POST_DATA (#PCDATA)>

<!ELEMENT RESPONSE (DATETIME, REPORT_LIST?)>
<!ELEMENT REPORT_LIST (REPORT+)>
<!ELEMENT REPORT (ID, TITLE, TYPE, USER_LOGIN, LAUNCH_DATETIME,
                  OUTPUT_FORMAT, SIZE, STATUS, EXPIRATION_DATETIME)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT TITLE (#PCDATA)>
<!ELEMENT TYPE (#PCDATA)>
<!ELEMENT USER_LOGIN (#PCDATA)>
<!ELEMENT LAUNCH_DATETIME (#PCDATA)>
<!ELEMENT OUTPUT_FORMAT (#PCDATA)>
<!ELEMENT SIZE (#PCDATA)>
<!ELEMENT STATUS (STATE, MESSAGE?, PERCENT?)>
<!ELEMENT EXPIRATION_DATETIME (#PCDATA)>
<!ELEMENT STATE (#PCDATA)>
<!ELEMENT MESSAGE (#PCDATA)>
<!ELEMENT PERCENT (#PCDATA)>
<!ELEMENT EXPIRATION_DATETIME (#PCDATA)>
<!-- EOF -->
```

XPaths for Report List Output

This section describes the XPaths for the report list output (report_list_output.dtd).

XPath	element specifications / notes
/REPORT_LIST_OUTPUT	(REQUEST?, RESPONSE)
/REPORT_LIST_OUTPUT/REQUEST	(DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?, POST_DATA?)
/REPORT_LIST_OUTPUT/REQUEST/DATETIME	(#PCDATA) The date and time of the request.
/REPORT_LIST_OUTPUT/REQUEST/USER_LOGIN	(#PCDATA) The user login ID of the user who made the request.
/REPORT_LIST_OUTPUT/REQUEST/RESOURCE	(#PCDATA) The resource specified for the request.
/REPORT_LIST_OUTPUT/REQUEST/PARAM_LIST	(PARAM+)
/REPORT_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM	(KEY, VALUE)
/REPORT_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM/KEY	(#PCDATA) The input parameter name.
/REPORT_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM/VALUE	(#PCDATA) The input parameter value.
/REPORT_LIST_OUTPUT/REQUEST/POST_DATA	(#PCDATA) The POST data, if any.
/REPORT_LIST_OUTPUT/RESPONSE	(DATETIME, REPORT_LIST?)
/REPORT_LIST_OUTPUT/RESPONSE/REPORT_LIST	(REPORT+)
/REPORT_LIST_OUTPUT/RESPONSE/REPORT_LIST/REPORT	(ID, TITLE, TYPE, USER_LOGIN, LAUNCH_DATETIME, OUTPUT_FORMAT, SIZE, STATUS, EXPIRATION_DATETIME)
/REPORT_LIST_OUTPUT/RESPONSE/REPORT_LIST/REPORT/ID	(#PCDATA) The report ID of the report.
/REPORT_LIST_OUTPUT/RESPONSE/REPORT_LIST/REPORT/TITLE	(#PCDATA) The report title.
/REPORT_LIST_OUTPUT/RESPONSE/REPORT_LIST/REPORT/TYPE	(#PCDATA) The report type: Map, Scan, Compliance, Remediation, Scorecard, WAS, Web Application Scorecard, or Patch.
/REPORT_LIST_OUTPUT/RESPONSE/REPORT_LIST/REPORT/USER_LOGIN	(#PCDATA) The user login ID of the user who launched the report.
/REPORT_LIST_OUTPUT/RESPONSE/REPORT_LIST/REPORT/LAUNCH_DATETIME	(#PCDATA) The date and time when the report was launched.

XPath	element specifications / notes
/REPORT_LIST_OUTPUT/RESPONSE/REPORT_LIST/REPORT/OUTPUT_FORMAT (#PCDATA)	
	The report output format: HTML, XML, PDF, MHT, CSV, or Online (for Qualys Patch Report only).
/REPORT_LIST_OUTPUT/RESPONSE/REPORT_LIST/REPORT/SIZE (#PCDATA)	
	The report size.
/REPORT_LIST_OUTPUT/RESPONSE/REPORT_LIST/REPORT/STATUS	
	(STATE, MESSAGE?, PERCENT?)
/REPORT_LIST_OUTPUT/RESPONSE/REPORT_LIST/REPORT/STATUS/STATE (#PCDATA)	
	The report state: Running, Finished, Canceled or Errors.
/REPORT_LIST_OUTPUT/RESPONSE/REPORT_LIST/REPORT/STATUS/MESSAGE (#PCDATA)	
	The report status message.
/REPORT_LIST_OUTPUT/RESPONSE/REPORT_LIST/REPORT/STATUS/PERCENT (#PCDATA)	
	For a report in progress, the percentage complete.
/REPORT_LIST_OUTPUT/RESPONSE/REPORT_LIST/REPORT/EXPIRATION_DATETIME (#PCDATA)	
	The report expiration date and time.

Network List

The network list XML output is returned from a network list API v2 request. The DTD can be found at the following URL (where <qualysapi.qualys.com> is the API server URL where your account is located):

```
https://qualysapi.qualys.com/network_list_output.dtd
```

DTD for Network List Output

A recent DTD for network list output (network_list_output.dtd) is below.

```
<!-- QUALYS NETWORK_LIST_OUTPUT DTD -->
<!ELEMENT NETWORK_LIST_OUTPUT (REQUEST?,RESPONSE)>
<!ELEMENT REQUEST (DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?,
                   POST_DATA?)>
<!ELEMENT DATETIME (#PCDATA)>
<!ELEMENT USER_LOGIN (#PCDATA)>
<!ELEMENT RESOURCE (#PCDATA)>
<!ELEMENT PARAM_LIST (PARAM+)>
<!ELEMENT PARAM (KEY, VALUE)>
<!ELEMENT KEY (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>
<!-- if returned, POST_DATA will be urlencoded -->
<!ELEMENT POST_DATA (#PCDATA)>
<!ELEMENT RESPONSE (DATETIME, NETWORK_LIST?)>
<!ELEMENT NETWORK_LIST (NETWORK+)>
<!ELEMENT NETWORK (ID, NAME, SCANNER_APPLIANCE_LIST?)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT NAME  (#PCDATA)>
<!ELEMENT SCANNER_APPLIANCE_LIST (SCANNER_APPLIANCE+)>
<!ELEMENT SCANNER_APPLIANCE (ID, FRIENDLY_NAME)>
<!ELEMENT FRIENDLY_NAME (#PCDATA)>
<!-- EOF -->
```

XPaths for Network List Output

This section describes the XPaths for network list output (network_list_output.dtd).

XPath	element specifications / notes
/NETWORK_LIST_OUTPUT	(REQUEST?, RESPONSE)
/NETWORK_LIST_OUTPUT/REQUEST	(DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?, POST_DATA?)
/NETWORK_LIST_OUTPUT/REQUEST/DATETIME	(#PCDATA) The date and time of the request.

XPath	element specifications / notes
/NETWORK_LIST_OUTPUT/REQUEST/USER_LOGIN (#PCDATA)	
	The user login ID of the user who made the request.
/NETWORK_LIST_OUTPUT/REQUEST/RESOURCE (#PCDATA)	
	The resource specified for the request.
/NETWORK_LIST_OUTPUT/REQUEST/PARAM_LIST (PARAM+)	
/NETWORK_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM (KEY, VALUE)	
/NETWORK_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM/KEY (#PCDATA)	
	The input parameter name.
/NETWORK_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM/VALUE (#PCDATA)	
	The input parameter value.
/NETWORK_LIST_OUTPUT/REQUEST/POST_DATA (#PCDATA)	
	The POST data.
/NETWORK_LIST_OUTPUT/RESPONSE (DATETIME, NETWORK_LIST?) /NETWORK_LIST_OUTPUT/RESPONSE/DATETIME (#PCDATA)	
	The date and time of the response.
/NETWORK_LIST_OUTPUT/RESPONSE/NETWORK_LIST (NETWORK+)	
/NETWORK_LIST_OUTPUT/RESPONSE/NETWORK_LIST/NETWORK (ID, NAME, SCANNER_APPLIANCE_LIST?)	
/NETWORK_LIST_OUTPUT/RESPONSE/NETWORK_LIST/NETWORK/ID (#PCDATA)	
	The network ID.
/NETWORK_LIST_OUTPUT/RESPONSE/NETWORK_LIST/NETWORK/NAME (#PCDATA)	
	The network name.
/NETWORK_LIST_OUTPUT/RESPONSE/NETWORK_LIST/NETWORK/ SCANNER_APPLIANCE_LIST (SCANNER_APPLIANCE+) /NETWORK_LIST_OUTPUT/RESPONSE/NETWORK_LIST/NETWORK/ SCANNER_APPLIANCE_LIST/SCANNER_APPLIANCE (ID, FRIENDLY_NAME) /NETWORK_LIST_OUTPUT/RESPONSE/NETWORK_LIST/NETWORK/ SCANNER_APPLIANCE_LIST/SCANNER_APPLIANCE/ID (#PCDATA)	
	The ID of a scanner appliance assigned to the network.
/NETWORK_LIST_OUTPUT/RESPONSE/NETWORK_LIST/NETWORK/ SCANNER_APPLIANCE_LIST/SCANNER_APPLIANCE/FRIENDLY_NAME (#PCDATA)	
	The name of a scanner appliance assigned to the network.

APPENDIX B

Asset XML

This appendix describes the XML output returned from API V2 requests for host and IP address data using the Asset API functions.

- IP List Output
- Host List Output
- Host List Detection Output
- Excluded Hosts List Output
- Excluded Hosts Change History Output
- Virtual Host List Output
- IPv6 Mapping Records List Output
- Restricted IPs List Output
- Asset Group List Output

IP List Output

The IP list output identifies IP addresses in the user account. The IP list XML is returned from a IP list API request.

The DTD can be found at the following URL:

`https://qualysapi.qualys.com/api/2.0/fo/asset/ip/ip_list_output.dtd`

where <qualysapi.qualys.com> is the API server URL where your account is located.

DTD for IP List Output

A recent DTD for the IP list output is shown below.

```
<!-- QUALYS IP_OUTPUT DTD -->
<!ELEMENT IP_LIST_OUTPUT (REQUEST?,RESPONSE)>

<!ELEMENT REQUEST (DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?,
                   POST_DATA?)>
<!ELEMENT DATETIME (#PCDATA)>
<!ELEMENT USER_LOGIN (#PCDATA)>
<!ELEMENT RESOURCE (#PCDATA)>
<!ELEMENT PARAM_LIST (PARAM+)>
<!ELEMENT PARAM (KEY, VALUE)>
<!ELEMENT KEY (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>
<!-- if returned, POST_DATA will be urlencoded -->
<!ELEMENT POST_DATA (#PCDATA)>

<!ELEMENT RESPONSE (DATETIME, IP_SET?)>

<!ELEMENT IP_SET ((IP|IP_RANGE)+)>
<!ELEMENT IP (#PCDATA)>
<!ELEMENT IP_RANGE (#PCDATA)>
<!-- EOF -->
```

XPaths for IP List Output

This section describes the XPaths for the IP list output.

XPath	element specifications / notes
/IP_LIST_OUTPUT	(REQUEST?, RESPONSE)
/IP_LIST_OUTPUT/REQUEST	(DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST, POST_DATA?)
/IP_LIST_OUTPUT/REQUEST/DATETIME	(#PCDATA)
	The date and time of the API request.
/IP_LIST_OUTPUT/REQUEST/USER_LOGIN	(#PCDATA)
	The user login of the user who made the request.
/IP_LIST_OUTPUT/REQUEST/RESOURCE	(#PCDATA)
	The resource specified for the request.
/IP_LIST_OUTPUT/REQUEST/PARAM_LIST	(PARAM+))
/IP_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM	(KEY, VALUE))
/IP_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM/KEY	(#PCDATA)
	The input parameter name.
/IP_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM/VALUE	(#PCDATA)
	The input parameter value.
/IP_LIST_OUTPUT/REQUEST/POST_DATA	(#PCDATA)
	The POST data, if any.
/IP_LIST_OUTPUT/RESPONSE	(DATETIME, IP_SET)
/IP_LIST_OUTPUT/RESPONSE/DATETIME	(#PCDATA)
	The date and time of the Qualys response.
/IP_LIST_OUTPUT/RESPONSE/IP_SET	((IP|IP_RANGE)+)
/IP_LIST_OUTPUT/RESPONSE/IP_SET/IP	(#PCDATA)
	An IP address.
/IP_LIST_OUTPUT/RESPONSE/IP_SET/IP_RANGE	(#PCDATA)
	An IP address range.

Host List Output

The host list XML output provides information about hosts in the user account that have been scanned. The host list output is returned from a host list API request.

The DTD can be found at the following URL:

`https://qualysapi.qualys.com/api/2.0/fo/asset/host/host_list_output.dtd`

where <qualysapi.qualys.com> is the API server URL where your account is located.

DTD for Host List Output

A recent DTD for the host list output is shown below.

```
<!-- QUALYS HOST_OUTPUT DTD -->

<!ELEMENT HOST_LIST_OUTPUT (REQUEST?,RESPONSE)>

<!ELEMENT REQUEST (DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?,
                   POST_DATA?)>
<!ELEMENT DATETIME (#PCDATA)>
<!ELEMENT USER_LOGIN (#PCDATA)>
<!ELEMENT RESOURCE (#PCDATA)>
<!ELEMENT PARAM_LIST (PARAM+)>
<!ELEMENT PARAM (KEY, VALUE)>
<!ELEMENT KEY (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>
<!-- if returned, POST_DATA will be urlencoded -->
<!ELEMENT POST_DATA (#PCDATA)>

<!ELEMENT RESPONSE (DATETIME, (HOST_LIST|ID_SET)?, WARNING?, GLOSSARY?)>
<!ELEMENT HOST_LIST (HOST+)>
<!ELEMENT HOST (ID, IP?, TRACKING_METHOD?, DNS?, EC2_INSTANCE_ID?,
                NETBIOS?, OS?, QG_HOSTID?, TAGS?,
                LAST_VULN_SCAN_DATETIME?, LAST_COMPLIANCE_SCAN_DATETIME?,
                OWNER?, COMMENTS?, USER_DEF?, ASSET_GROUP_IDS?)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT IP (#PCDATA)>
<!ELEMENT TRACKING_METHOD (#PCDATA)>
<!ELEMENT DNS (#PCDATA)>
<!ELEMENT EC2_INSTANCE_ID (#PCDATA)>
<!ELEMENT NETBIOS (#PCDATA)>
<!ELEMENT OS (#PCDATA)>
<!ELEMENT QG_HOSTID (#PCDATA)>
<!ELEMENT TAGS (TAG+)>
<!ELEMENT TAG (TAG_ID?, NAME?)>
```

```
<!ELEMENT TAG_ID (#PCDATA)>
<!ELEMENT NAME (#PCDATA)>
<!ELEMENT LAST_VULN_SCAN_DATETIME (#PCDATA)>
<!ELEMENT LAST_COMPLIANCE_SCAN_DATETIME (#PCDATA)>
<!ELEMENT OWNER (#PCDATA)>
<!ELEMENT COMMENTS (#PCDATA)>
<!ELEMENT USER_DEF (LABEL_1?, LABEL_2?, LABEL_3?, VALUE_1?, VALUE_2?,
                    VALUE_3?)>
<!ELEMENT LABEL_1 (#PCDATA)>
<!ELEMENT LABEL_2 (#PCDATA)>
<!ELEMENT LABEL_3 (#PCDATA)>
<!ELEMENT VALUE_1 (#PCDATA)>
<!ELEMENT VALUE_2 (#PCDATA)>
<!ELEMENT VALUE_3 (#PCDATA)>

<!ELEMENT ASSET_GROUP_IDS (#PCDATA)>

<!ELEMENT ID_SET ((ID|ID_RANGE)+)>
<!ELEMENT ID_RANGE (#PCDATA)>

<!ELEMENT WARNING (CODE?, TEXT, URL?)>
<!ELEMENT CODE (#PCDATA)>
<!ELEMENT TEXT (#PCDATA)>
<!ELEMENT URL (#PCDATA)>

<!ELEMENT GLOSSARY (USER_DEF?, USER_LIST?, ASSET_GROUP_LIST?)>

<!ELEMENT USER_LIST (USER+)>
<!ELEMENT USER (USER_LOGIN, FIRST_NAME, LAST_NAME)>
<!ELEMENT FIRST_NAME (#PCDATA)>
<!ELEMENT LAST_NAME (#PCDATA)>

<!ELEMENT ASSET_GROUP_LIST (ASSET_GROUP+)>
<!ELEMENT ASSET_GROUP (ID, TITLE)>
<!ELEMENT TITLE (#PCDATA)>
<!-- EOF -->
```

XPaths for Host List Output

This section describes the XPaths for the host list output.

XPath	element specifications / notes
/HOST_OUTPUT	(REQUEST?,RESPONSE)
/HOST_OUTPUT/REQUEST	(DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?, POST_DATA?)
/HOST_OUTPUT/REQUEST/DATETIME	(#PCDATA)
	The date and time of the API request.
/HOST_OUTPUT/REQUEST/USER_LOGIN	(#PCDATA)
	The user login ID of the user who made the request.
/HOST_OUTPUT/REQUEST/RESOURCE	(#PCDATA)
	The resource specified for the request.
/HOST_OUTPUT/REQUEST/PARAM_LIST	(PARAM+))
/HOST_OUTPUT/REQUEST/PARAM_LIST/PARAM	(KEY, VALUE))
/HOST_OUTPUT/REQUEST/PARAM_LIST/PARAM/KEY	(#PCDATA)
	An input parameter name.
/HOST_OUTPUT/REQUEST/PARAM_LIST/PARAM/VALUE	(#PCDATA)
	An input parameter value.
/HOST_OUTPUT/REQUEST/POST_DATA	(#PCDATA)
	The POST data, if any.
/HOST_OUTPUT/RESPONSE	(DATETIME, (HOST_LIST|ID_SET)?, WARNING?, GLOSSARY?)
/HOST_OUTPUT/RESPONSE/DATETIME	(#PCDATA)
	The date and time of the Qualys response.
/HOST_OUTPUT/RESPONSE/HOST_LIST	(HOST+)
/HOST_OUTPUT/RESPONSE/HOST_LIST/HOST	(ID, IP?, TRACKING_METHOD?, DNS?, EC2_INSTANCE_ID, NETBIOS?, OS?, QG_HOSTID?, TAGS?, LAST_VULN_SCAN_DATETIME?, LAST_COMPLIANCE_SCAN_DATETIME?, OWNER?, COMMENTS?, USER_DEF?, ASSET_GROUP_IDS?)
	The HOST element is returned when the “details” input parameter is set to “basic” or “all” or if the parameter is unspecified.
/HOST_OUTPUT/RESPONSE/HOST_LIST/HOST/ID	(#PCDATA)
	The host ID.
/HOST_OUTPUT/RESPONSE/HOST_LIST/HOST/IP	(#PCDATA)
	The host IP address.
/HOST_OUTPUT/RESPONSE/HOST_LIST/HOST/TRACKING_METHOD	(#PCDATA)
	The tracking method assigned to the host: IP, DNS, or NETBIOS.
/HOST_OUTPUT/RESPONSE/HOST_LIST/HOST/DNS	(#PCDATA)
	The DNS host name.

XPath	element specifications / notes
/HOST_OUTPUT/RESPONSE/HOST_LIST/NETBIOS (#PCDATA)	
	The NetBIOS host name.
/HOST_OUTPUT/RESPONSE/HOST_LIST/EC2_INSTANCE_ID (#PCDATA)	
	The EC2 host instance ID.
/HOST_OUTPUT/RESPONSE/HOST_LIST/HOST/OS (#PCDATA)	
	The operating system detected on the host.
/HOST_OUTPUT/RESPONSE/HOST_LIST/HOST/QG_HOSTID (#PCDATA)	
	The QualsysGuard host ID assigned to the host when Agentless Tracking is used.
/HOST_OUTPUT/RESPONSE/HOST_LIST/HOST/TAGS (TAG_ID?, NAME?)	
/HOST_OUTPUT/RESPONSE/HOST_LIST/HOST/TAGS/TAG_ID (#PCDATA)	
	A tag ID associated with the host when show_tags=1 is specified.
/HOST_OUTPUT/RESPONSE/HOST_LIST/HOST/TAGS/NAME (#PCDATA)	
	A tag name associated with the host when show_tags=1 is specified.
/HOST_OUTPUT/RESPONSE/HOST_LIST/HOST/LAST_SCAN_DATETIME (#PCDATA)	
	The date and time of the most recent vulnerability scan.
/HOST_OUTPUT/RESPONSE/HOST_LIST/HOST/LAST_COMPLIANCE_SCAN_DATETIME (#PCDATA)	
	The date and time of the most recent compliance scan.
/HOST_OUTPUT/RESPONSE/HOST_LIST/HOST/OWNER (#PCDATA)	
	The asset owner.
/HOST_OUTPUT/RESPONSE/HOST_LIST/HOST/COMMENTS (#PCDATA)	
	The comments defined for the host.
/HOST_OUTPUT/RESPONSE/HOST_LIST/HOST/USER_DEF	
	(LABEL_1?, LABEL_2?, LABEL_3?, VALUE_1?, VALUE_2?, VALUE_3?)
	A set of host attributes assigned to the host. Three user-defined attributes are defined for the subscription.
/HOST_OUTPUT/RESPONSE/HOST_LIST/HOST/USER_DEF/LABEL_n (#PCDATA)	
	Not returned inside the <HOST> element. Returned inside <GLOSSARY>.
/HOST_OUTPUT/RESPONSE/HOST_LIST/HOST/USER_DEF/VALUE_n (#PCDATA)	
	A host attribute value. Three elements are returned, one element for each of the three values. The elements are: <VALUE_1>, <VALUE_2> and <VALUE_3>.
/HOST_OUTPUT/RESPONSE/HOST_LIST/HOST/ASSET_GROUP_IDS (#PCDATA)	
	The asset group IDs for the asset groups which the host belongs to.
/HOST_OUTPUT/RESPONSE/ID_SET ((ID|ID_RANGE)+)	
	The ID_SET element is returned when the “details” input parameter is set to “none”.
/HOST_OUTPUT/RESPONSE/ID_SET/ID (#PCDATA)	
	A host ID.
/HOST_OUTPUT/RESPONSE/ID_SET/ID_RANGE (#PCDATA)	
	A host ID range.

XPath	element specifications / notes
/HOST_OUTPUT/RESPONSE/WARNING (CODE?, TEXT, URL?)	
/HOST_OUTPUT/RESPONSE/WARNING/CODE (#PCDATA)	
	The warning code. This code appears when the API request identifies more than 1,000 records (hosts) or the custom truncation limit.
/HOST_OUTPUT/RESPONSE/WARNING/TEXT (#PCDATA)	
	The warning message text. This message appears when the API request identifies more than 1,000 records (hosts) or the custom truncation limit.
/HOST_OUTPUT/RESPONSE/WARNING/URL (#PCDATA)	
	The URL for making another request for the next batch of host records.
/HOST_OUTPUT/RESPONSE/GLOSSARY (USER_DEF?, USER_LIST?, ASSET_GROUP_LIST?)	
/HOST_OUTPUT/RESPONSE/GLOSSARY/USER_DEF (#PCDATA)	
	(LABEL_1?, LABEL_2?, LABEL_3?, VALUE_1?, VALUE_2?, VALUE_3?)
	A set of host attributes assigned to the host. Three user-defined attributes are defined for the subscription.
/HOST_OUTPUT/RESPONSE/GLOSSARY/USER_DEF/LABEL_n (#PCDATA)	
	A host attribute label, as defined for the subscription. When the default labels are used the elements are: <LABEL_1>Location, <LABEL_2>Function and <LABEL_3>Asset Tag. The labels may be customized within Qualys.
/HOST_OUTPUT/RESPONSE/GLOSSARY/USER_DEF/VALUE_n (#PCDATA)	
	Not returned inside the <GLOSSARY> element. Returned inside <HOST>.
/HOST_OUTPUT/RESPONSE/GLOSSARY/USER_LIST (USER+)	
	A list of users who are asset owners for the hosts in the host list output.
/HOST_OUTPUT/RESPONSE/GLOSSARY/USER_LIST/USER (USER_LOGIN, FIRST_NAME, LAST_NAME)	
	A user who is an asset owner for a host in the host list output.
/HOST_OUTPUT/RESPONSE/GLOSSARY/USER_LIST/USER_LOGIN (#PCDATA)	
	A user login ID.
/HOST_OUTPUT/RESPONSE/GLOSSARY/USER_LIST/USER/FIRST_NAME (#PCDATA)	
	A user's first name.
/HOST_OUTPUT/RESPONSE/GLOSSARY/USER_LIST/LAST_NAME	
	A user's last name.
/HOST_OUTPUT/RESPONSE/GLOSSARY/ASSET_GROUP_LIST (ASSET_GROUP+)	
	A list of asset groups which hosts in the host list output belong to.
/HOST_OUTPUT/RESPONSE/GLOSSARY/ASSET_GROUP_LIST/ASSET_GROUP (ID, TITLE)	
/HOST_OUTPUT/RESPONSE/GLOSSARY/ASSET_GROUP_LIST/ASSET_GROUP/ID	
	An asset group ID.
/HOST_OUTPUT/RESPONSE/GLOSSARY/ASSET_GROUP_LIST/ASSET_GROUP/TITLE	
	An asset group title.

Host List Detection Output

The host list detection XML output provides information about VM scanned hosts in the user account, including "automatic" data. The host list detection output is returned from a host list detection API request.

The DTD can be found at the following URL:

```
https://qualysapi.qualys.com/api/2.0/fo/asset/host/vm/detection
/host_list_vm_detection_output.dtd
```

where <qualysapi.qualys.com> is the API server URL where your account is located.

DTD for Host List Output

A recent DTD for the host list detection output is shown below.

```
<!-- QUALYS HOST_LIST_VM_DETECTION_OUTPUT DTD -->

<!ELEMENT HOST_LIST_VM_DETECTION_OUTPUT (REQUEST?,RESPONSE)>

<!ELEMENT REQUEST (DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?,
                   POST_DATA?)>
<!ELEMENT DATETIME (#PCDATA)>
<!ELEMENT USER_LOGIN (#PCDATA)>
<!ELEMENT RESOURCE (#PCDATA)>
<!ELEMENT PARAM_LIST (PARAM+)>
<!ELEMENT PARAM (KEY, VALUE)>
<!ELEMENT KEY (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>
<!-- if returned, POST_DATA will be urlencoded -->
<!ELEMENT POST_DATA (#PCDATA)>

<!ELEMENT RESPONSE (DATETIME, HOST_LIST?, WARNING?)>
<!ELEMENT HOST_LIST (HOST+)>
<!ELEMENT HOST (ID, IP?, IPV6?, TRACKING_METHOD?, OS?, OS_CPE?, DNS?,
                NETBIOS?, QG_HOSTID?,
                LAST_SCAN_DATETIME?, TAGS?, DETECTION_LIST)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT IP (#PCDATA)>
<!ELEMENT IPV6 (#PCDATA)>
<!ELEMENT TRACKING_METHOD (#PCDATA)>
<!ELEMENT OS (#PCDATA)>
<!ELEMENT OS_CPE (#PCDATA)>
<!ELEMENT DNS (#PCDATA)>
<!ELEMENT NETBIOS (#PCDATA)>
<!ELEMENT QG_HOSTID (#PCDATA)>
```

```
<!ELEMENT LAST_SCAN_DATETIME (#PCDATA)>
<!ELEMENT TAGS (TAG+)>
<!ELEMENT TAG (TAG_ID?, NAME, COLOR?, BACKGROUND_COLOR?)>
<!ELEMENT TAG_ID (#PCDATA)>
<!ELEMENT NAME (#PCDATA)>
<!ELEMENT COLOR (#PCDATA)>
<!ELEMENT BACKGROUND_COLOR (#PCDATA)>
<!ELEMENT DETECTION_LIST (DETECTION+)>
<!ELEMENT DETECTION (QID, TYPE, SEVERITY?, PORT?, PROTOCOL?, FQDN?, SSL?,
                     INSTANCE?, RESULTS?, STATUS?, FIRST_FOUND_DATETIME?,
                     LAST_FOUND_DATETIME?, LAST_TEST_DATETIME?,
                     LAST_UPDATE_DATETIME?, LAST_FIXED_DATETIME?)>
<!ELEMENT QID (#PCDATA)>
<!ELEMENT TYPE (#PCDATA)>
<!ELEMENT SEVERITY (#PCDATA)>
<!ELEMENT PORT (#PCDATA)>
<!ELEMENT PROTOCOL (#PCDATA)>
<!ELEMENT FQDN (#PCDATA)>
<!ELEMENT SSL (#PCDATA)>
<!ELEMENT INSTANCE (#PCDATA)>
<!ELEMENT RESULTS (#PCDATA)>
<!ELEMENT STATUS (#PCDATA)>
<!ELEMENT FIRST_FOUND_DATETIME (#PCDATA)>
<!ELEMENT LAST_FOUND_DATETIME (#PCDATA)>
<!ELEMENT LAST_TEST_DATETIME (#PCDATA)>
<!ELEMENT LAST_UPDATE_DATETIME (#PCDATA)>
<!ELEMENT LAST_FIXED_DATETIME (#PCDATA)>
<!ELEMENT WARNING (CODE?, TEXT, URL?)>
<!ELEMENT CODE (#PCDATA)>
<!ELEMENT TEXT (#PCDATA)>
<!ELEMENT URL (#PCDATA)>
<!-- EOF -->
```

XPaths for Host List Detection Output

This section describes the XPaths for the host list detection output.

XPath	element specifications / notes
/HOST_LIST_VM_DETECTION_OUTPUT	
	(REQUEST?,RESPONSE)
/HOST_LIST_VM_DETECTION_OUTPUT/REQUEST	
	(DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?, POST_DATA?)
/HOST_LIST_VM_DETECTION_OUTPUT/REQUEST/DATETIME	(#PCDATA)
	The date and time of the API request.
/HOST_LIST_VM_DETECTION_OUTPUT/REQUEST/USER_LOGIN	(#PCDATA)
	The user login ID of the user who made the request.

XPath	element specifications / notes
/HOST_LIST_VM_DETECTION_OUTPUT/REQUEST/RESOURCE (#PCDATA)	
	The resource specified for the request.
/HOST_LIST_VM_DETECTION_OUTPUT/REQUEST/PARAM_LIST (PARAM+))	
/HOST_LIST_VM_DETECTION_OUTPUT/REQUEST/PARAM_LIST/PARAM (KEY, VALUE))	
/HOST_LIST_VM_DETECTION_OUTPUT/REQUEST/PARAM_LIST/PARAM/KEY (#PCDATA)	
	An input parameter name.
/HOST_LIST_VM_DETECTION_OUTPUT/REQUEST/PARAM_LIST/PARAM/VALUE (#PCDATA)	
	An input parameter value.
/HOST_LIST_VM_DETECTION_OUTPUT/REQUEST/POST_DATA (#PCDATA)	
	The POST data, if any.
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE	
	(DATETIME, HOST_LIST?, WARNING?)
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE/DATETIME (#PCDATA)	
	The date and time of the Qualys response.
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE/HOST_LIST (HOST+)	
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE/HOST_LIST/HOST	
	(ID, IP?, IPV6?, TRACKING_METHOD?, OS?, OS_CPE?, DNS?, NETBIOS?, QG_HOSTID?, LAST_SCAN_DATETIME?, TAGS?, DETECTION_LIST)
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE/HOST_LIST/HOST/ID (#PCDATA)	
	The host ID.
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE/HOST_LIST/HOST/IP (#PCDATA)	
	The host IPv4 address.
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE/HOST_LIST/HOST/IP (#PCDATA)	
	The host IPv6 address. This appears only if the IPv6 feature is enabled for the subscription and the IPv6 address is defined in an IPv4 to IPv6 mapping record.
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE/HOST_LIST/HOST/TRACKING_METHOD (#PCDATA)	
	The tracking method assigned to the host: IP, DNS, or NETBIOS.
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE/HOST_LIST/HOST/OS (#PCDATA)	
	The operating system detected on the host.
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE/HOST_LIST/HOST/OS_CPE (#PCDATA)	
	The OS CPE name assigned to the operating system detected on the host. (The OS CPE name appears only when the OS CPE feature is enabled for the subscription, and an authenticated scan was run on this host after enabling this feature.)
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE/HOST_LIST/HOST/DNS (#PCDATA)	
	The host's DNS name.
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE/HOST_LIST/NETBIOS (#PCDATA)	
	The host's NetBIOS name.

XPath	element specifications / notes
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE/HOST_LIST/QG_HOSTID (#PCDATA)	
	The Qualys host ID assigned to the host when agentless tracking is used.
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE/HOST_LIST/HOST/LAST_SCAN_DATETIME (#PCDATA)	
	The date and time of the most recent vulnerability scan of the host.
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE/HOST_LIST/HOST/TAGS (TAG+)	
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE/HOST_LIST/HOST/TAGS/TAG (TAG_ID?, NAME, COLOR?, BACKGROUND_COLOR?)	
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE/HOST_LIST/HOST/TAGS/TAG/TAG_ID (#PCDATA)	
	The ID of a tag associated with the host when show_tags=1 is specified.
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE/HOST_LIST/HOST/TAGS/TAG/NAME (#PCDATA)	
	The name of a tag associated with the host when show_tags=1 is specified.
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE/HOST_LIST/HOST/TAGS/TAG/COLOR (#PCDATA)	
	The color of a tag associated with the host when show_tags=1 is specified.
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE/HOST_LIST/HOST/TAGS/TAG/ BACKGROUND_COLOR (#PCDATA)	
	The background color of a tag associated with the host when show_tags=1 is specified.
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE/HOST_LIST/HOST/DETECTION_LIST (DETECTION+))	
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE/HOST_LIST/HOST/DETECTION_LIST/DETECTION	
	(QID, TYPE, SEVERITY?, PORT?, PROTOCOL?, FQDN?, SSL?, INSTANCE?, RESULTS?, STATUS?, FIRST_FOUND_DATETIME?, LAST_FOUND_DATETIME?, LAST_TEST_DATETIME?, LAST_UPDATE_DATETIME?, LAST_FIXED_DATETIME?)
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE/HOST_LIST/HOST/DETECTION_LIST/ DETECTION/QID (#PCDATA)	
	The QID for the vulnerability in the detection record.
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE/HOST_LIST/HOST/DETECTION_LIST/ DETECTION/TYPE (#PCDATA)	
	The type of vulnerability in the detection record: Confirmed for a confirmed vulnerability, Potential for a potential vulnerability, and Info for an information gathered.
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE/HOST_LIST/HOST/DETECTION_LIST/ DETECTION/SEVERITY (#PCDATA)	
	The severity of the vulnerability.
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE/HOST_LIST/HOST/DETECTION_LIST/ DETECTION/PORT (#PCDATA)	
	The port number that the vulnerability was detected on.
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE/HOST_LIST/HOST/DETECTION_LIST/ DETECTION/PROTOCOL (#PCDATA)	
	The protocol the vulnerability was detected on.

XPath	element specifications / notes
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE/HOST_LIST/HOST/DETECTION_LIST/DETECTION/FQDN (#PCDATA)	
	The Fully Qualified Domain Name (FQDN) of the host.
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE/HOST_LIST/HOST/DETECTION_LIST/DETECTION/SSL (#PCDATA)	
	The value 1 is returned if the vulnerability was detected over SSL. The value 0 is returned if the vulnerability was not detected over SSL. This element is not returned for information gathered.
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE/HOST_LIST/HOST/DETECTION_LIST/DETECTION/INSTANCE (#PCDATA)	
	The Oracle DB instance the vulnerability was detected on.
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE/HOST_LIST/HOST/DETECTION_LIST/DETECTION/RESULTS (#PCDATA)	
	The scan test results, if any, returned by the service for the detection record.
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE/HOST_LIST/HOST/DETECTION_LIST/DETECTION/STATUS (#PCDATA)	
	The current vulnerability status of the vulnerability in the detection record.
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE/HOST_LIST/HOST/DETECTION_LIST/DETECTION/FIRST_FOUND_DATETIME (#PCDATA)	
	The date/time when the vulnerability was first found.
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE/HOST_LIST/HOST/DETECTION_LIST/DETECTION/LAST_FOUND_DATETIME (#PCDATA)	
	The most recent date/time when the vulnerability was found.
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE/HOST_LIST/HOST/DETECTION_LIST/DETECTION/LAST_TEST_DATETIME (#PCDATA)	
	The most recent date/time when the vulnerability was tested.
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE/HOST_LIST/HOST/DETECTION_LIST/DETECTION/LAST_UPDATE_DATETIME (#PCDATA)	
	The most recent date/time when the detection record was updated.
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE/HOST_LIST/HOST/DETECTION_LIST/DETECTION/LAST_FIXED_DATETIME (#PCDATA)	
	The date/time when the vulnerability was verified fixed by a scan.
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE/WARNING (CODE?, TEXT, URL?)	
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE/WARNING/CODE (#PCDATA)	
	The warning code. This code appears when the API request identifies more than 1,000 host records.
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE/WARNING/TEXT (#PCDATA)	
	The warning message text. This message appears when the API request identifies more than 1,000 host records.
/HOST_LIST_VM_DETECTION_OUTPUT/RESPONSE/WARNING/URL (#PCDATA)	
	The URL for making another request for the next batch of host records.

Excluded Hosts List Output

The excluded hosts list XML output provides information about excluded hosts in the user's account. The excluded host list output is returned from an excluded host list API request.

The DTD can be found at the following URL:

```
https://qualysapi.qualys.com/api/2.0/fo/asset/excluded_ip/
ip_list_output.dtd
```

where <qualysapi.qualys.com> is the API server URL where your account is located.

DTD for Excluded Host List Output

A recent DTD for the excluded hosts list output is shown below.

```
<!-- QUALYS IP_OUTPUT DTD -->

<!ELEMENT IP_LIST_OUTPUT (REQUEST?,RESPONSE)>

<!ELEMENT REQUEST (DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?,
                   POST_DATA?)>
<!ELEMENT DATETIME (#PCDATA)>
<!ELEMENT USER_LOGIN (#PCDATA)>
<!ELEMENT RESOURCE (#PCDATA)>
<!ELEMENT PARAM_LIST (PARAM+)>
<!ELEMENT PARAM (KEY, VALUE)>
<!ELEMENT KEY (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>
<!-- if returned, POST_DATA will be urlencoded -->
<!ELEMENT POST_DATA (#PCDATA)>

<!ELEMENT RESPONSE (DATETIME, IP_SET?)>

<!ELEMENT IP_SET ((IP|IP_RANGE)+)>
<!ELEMENT IP (#PCDATA)>
<!ATTLIST IP network_id  CDATA  #IMPLIED>
<!ATTLIST IP expiration_date  CDATA  #IMPLIED>
<!ELEMENT IP_RANGE (#PCDATA)>
<!ATTLIST IP_RANGE
  network_id  CDATA  #IMPLIED
  expiration_date  CDATA  #IMPLIED
>
<!-- EOF -->
```

XPaths for Excluded Hosts List Output

This section describes the XPaths for the excluded hosts list output.

XPath	element specifications / notes
/IP_LIST_OUTPUT	(REQUEST?, RESPONSE)
/IP_LIST_OUTPUT/REQUEST	(DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST, POST_DATA?)
/IP_LIST_OUTPUT/REQUEST/DATETIME	(#PCDATA)
	The date and time of the API request.
/IP_LIST_OUTPUT/REQUEST/USER_LOGIN	(#PCDATA)
	The user login of the user who made the request.
/IP_LIST_OUTPUT/REQUEST/RESOURCE	(#PCDATA)
	The resource specified for the request.
/IP_LIST_OUTPUT/REQUEST/PARAM_LIST	(PARAM+))
/IP_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM	(KEY, VALUE))
/IP_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM/KEY	(#PCDATA)
	The input parameter name.
/IP_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM/VALUE	(#PCDATA)
	The input parameter value.
/IP_LIST_OUTPUT/REQUEST/POST_DATA	(#PCDATA)
	The POST data, if any.
/IP_LIST_OUTPUT/RESPONSE	(DATETIME, IP_SET)
/IP_LIST_OUTPUT/RESPONSE/DATETIME	(#PCDATA)
	The date and time of the Qualys response.
/IP_LIST_OUTPUT/RESPONSE/IP_SET	((IP|IP_RANGE)+)
/IP_LIST_OUTPUT/RESPONSE/IP_SET/IP	(#PCDATA)
	An IP address, identifying an excluded host. If the Networks feature is enabled in your subscription, the attribute “network_id” is the network ID associated with this IP address. If an expiration date was specified when this IP was added to the list, the attribute “expiration_date” is the date when the IP will be removed from the list.
/IP_LIST_OUTPUT/RESPONSE/IP_SET/IP_RANGE	(#PCDATA)
	An IP address range, identifying excluded hosts. If the Networks feature is enabled in your subscription, the attribute “network_id” is the network ID associated with this IP range. If an expiration date was specified when this IP range was added to the list, the attribute “expiration_date” is the date when the IP range will be removed from the list.

Excluded Hosts Change History Output

The excluded hosts change history XML output provides information about excluded hosts change history in the user's account. The excluded host change history output is returned from an excluded host change history API request.

The DTD can be found at the following URL:

```
https://qualysapi.qualys.com/api/2.0/fo/asset/excluded_ip/history/
history_list_output.dtd
```

where <qualysapi.qualys.com> is the API server URL where your account is located.

DTD for Excluded Host Change History Output

A recent DTD for the excluded hosts change history output is shown below.

```
<!-- QUALYS HISTORY_LIST_OUTPUT DTD -->

<!ELEMENT HISTORY_LIST_OUTPUT (REQUEST?,RESPONSE)>

<!ELEMENT REQUEST (DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?,
                   POST_DATA?)>
<!ELEMENT DATETIME (#PCDATA)>
<!ELEMENT USER_LOGIN (#PCDATA)>
<!ELEMENT RESOURCE (#PCDATA)>
<!ELEMENT PARAM_LIST (PARAM+)>
<!ELEMENT PARAM (KEY, VALUE)>
<!ELEMENT KEY (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>
<!-- if returned, POST_DATA will be urlencoded -->
<!ELEMENT POST_DATA (#PCDATA)>

<!ELEMENT RESPONSE (DATETIME, HISTORY_LIST?, WARNING?, GLOSSARY?)>
<!ELEMENT HISTORY_LIST (HISTORY+)>
<!ELEMENT HISTORY (ID, IP_SET, ACTION, DATETIME, USER_LOGIN, COMMENTS)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT IP_SET ((IP|IP_RANGE)+)>
<!ELEMENT IP (#PCDATA)>
<!ELEMENT IP_RANGE (#PCDATA)>
<!ELEMENT ACTION (#PCDATA)>
<!ELEMENT COMMENTS (#PCDATA)>

<!ELEMENT WARNING (CODE?, TEXT, URL?)>
<!ELEMENT CODE (#PCDATA)>
<!ELEMENT TEXT (#PCDATA)>
<!ELEMENT URL (#PCDATA)>
```

```
<!ELEMENT GLOSSARY (USER_LIST)>
<!ELEMENT USER_LIST (USER+)>
<!ELEMENT USER (USER_LOGIN, FIRST_NAME, LAST_NAME, ROLE)>
<!ELEMENT FIRST_NAME (#PCDATA)>
<!ELEMENT LAST_NAME (#PCDATA)>
<!ELEMENT ROLE (#PCDATA)>
<!-- EOF -->
```

XPaths for Excluded Hosts Change History Output

This section describes the XPaths for the excluded hosts change history output.

XPath	element specifications / notes
/HISTORY_LIST_OUTPUT	(REQUEST?, RESPONSE)
/HISTORY_LIST_OUTPUT /REQUEST	(DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST, POST_DATA?)
/HISTORY_LIST_OUTPUT /REQUEST/DATETIME (#PCDATA)	The date and time of the API request.
/HISTORY_LIST_OUTPUT /REQUEST/USER_LOGIN (#PCDATA)	The user login of the user who made the request.
/HISTORY_LIST_OUTPUT /REQUEST/RESOURCE (#PCDATA)	The resource specified for the request.
/HISTORY_LIST_OUTPUT /REQUEST/PARAM_LIST (PARAM+))	
/HISTORY_LIST_OUTPUT /REQUEST/PARAM_LIST/PARAM (KEY, VALUE))	
/HISTORY_LIST_OUTPUT /REQUEST/PARAM_LIST/PARAM/KEY (#PCDATA)	The input parameter name.
/HISTORY_LIST_OUTPUT /REQUEST/PARAM_LIST/PARAM/VALUE (#PCDATA)	The input parameter value.
/HISTORY_LIST_OUTPUT /REQUEST/POST_DATA (#PCDATA)	The POST data, if any.
/HISTORY_LIST_OUTPUT/RESPONSE (DATETIME, HISTORY_LIST? WARNING?, GLOSSARY?)	
/HISTORY_LIST_OUTPUT/RESPONSE /DATETIME (#PCDATA)	The date and time of the Qualys response.
/HISTORY_LIST_OUTPUT/RESPONSE /HISTORY_LIST (HISTORY+)	
/HISTORY_LIST_OUTPUT/RESPONSE /HISTORY_LIST/HISTORY	(ID, IP_SET, ACTION, DATETIME, USER_LOGIN, COMMENTS))
/HISTORY_LIST_OUTPUT/RESPONSE /HISTORY_LIST/HISTORY/ID (#PCDATA)	An ID for an excluded hosts change history record.

XPath	element specifications / notes
/HISTORY_LIST_OUTPUT/RESPONSE /HISTORY_LIST/HISTORY/IP_SET	((IP, IP_RANGE)+)
/HISTORY_LIST_OUTPUT/RESPONSE /HISTORY_LIST/HISTORY/IP_SET/IP	(#PCDATA)
	An IP address range, identifying excluded hosts.
/HISTORY_LIST_OUTPUT/RESPONSE /HISTORY_LIST/HISTORY/IP_SET/RANGE	(#PCDATA)
	An IP address range, identifying excluded hosts.
/HISTORY_LIST_OUTPUT/RESPONSE /HISTORY_LIST/HISTORY/ACTION	(#PCDATA)
	An action associated with the change: Added for added excluded hosts, or Removed for removed excluded hosts.
/HISTORY_LIST_OUTPUT/RESPONSE /HISTORY_LIST/HISTORY/COMMENTS	(#PCDATA)
	User comments entered during the action associated with excluded hosts.
/HISTORY_LIST_OUTPUT /RESPONSE/WARNING	(CODE?, TEXT, URL?)
/HISTORY_LIST_OUTPUT /RESPONSE/WARNING/CODE	(#PCDATA)
	The warning code. This code appears when the API request identifies more than 1,000 excluded hosts change history records.
/HISTORY_LIST_OUTPUT /RESPONSE/WARNING/TEXT	(#PCDATA)
	The warning message text. This message appears when the API request identifies more than 1,000 excluded hosts change history records.
/HISTORY_LIST_OUTPUT /RESPONSE/WARNING/TEXT/URL	(#PCDATA)
	The URL for making another request for the next batch of excluded hosts change history records. The URL includes the “id_max” parameter for change history records with an ID less than or equal to a specified ID.
/HISTORY_LIST_OUTPUT /RESPONSE/GLOSSARY	(USER_LIST)
/HISTORY_LIST_OUTPUT /RESPONSE/GLOSSARY/USER_LIST	(USER+)
/HISTORY_LIST_OUTPUT /RESPONSE/GLOSSARY/USER_LIST/USER	(USER_LOGIN, FIRST_NAME, LAST_NAME, ROLE)
/HISTORY_LIST_OUTPUT /RESPONSE/GLOSSARY/USER_LIST/USER/FIRST_NAME	(#PCDATA)
	The first name of a user who performed an action on excluded hosts included in the XML output.
/HISTORY_LIST_OUTPUT /RESPONSE/GLOSSARY/USER_LIST/USER/LAST_NAME	(#PCDATA)
	The last name of a user who performed an action on excluded hosts included in the XML output.
/HISTORY_LIST_OUTPUT /RESPONSE/GLOSSARY/USER_LIST/USER/ROLE	(#PCDATA)
	The role of a user who performed an action on excluded hosts included in the XML output.

Virtual Host List Output

The virtual host list output identifies virtual hosts in the user account.

The DTD can be found at the following URL:

```
https://qualysapi.qualys.com/api/2.0/fo/asset/vhost/
vhost_list_output.dtd
```

where <qualysapi.qualys.com> is the API server URL where your account is located.

DTD for Virtual Host List Output

A recent DTD for the virtual host list output is shown below.

```
<!-- QUALYS VIRTUAL_HOST_OUTPUT DTD -->

<!ELEMENT VIRTUAL_HOST_LIST_OUTPUT (REQUEST?,RESPONSE)>

<!ELEMENT REQUEST (DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?,
                   POST_DATA?)>
<!ELEMENT DATETIME (#PCDATA)>
<!ELEMENT USER_LOGIN (#PCDATA)>
<!ELEMENT RESOURCE (#PCDATA)>
<!ELEMENT PARAM_LIST (PARAM+)>
<!ELEMENT PARAM (KEY, VALUE)>
<!ELEMENT KEY (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>
<!-- if returned, POST_DATA will be urlencoded -->
<!ELEMENT POST_DATA (#PCDATA)>

<!ELEMENT RESPONSE (DATETIME, (VIRTUAL_HOST_LIST)?, WARNING?)>
<!ELEMENT VIRTUAL_HOST_LIST (VIRTUAL_HOST+)>
<!ELEMENT VIRTUAL_HOST (IP, PORT, FQDN+)>
<!ELEMENT IP (#PCDATA)>
<!ELEMENT PORT (#PCDATA)>
<!ELEMENT FQDN (#PCDATA)>
```

XPaths for Virtual Host List Output

This section describes the XPaths for the virtual host list output.

XPath	element specifications / notes
/VIRTUAL_HOST_LIST_OUTPUT	(REQUEST?,RESPONSE)
/VIRTUAL_HOST_LIST_OUTPUT/REQUEST	(DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?, POST_DATA?)
/VIRTUAL_HOST_LIST_OUTPUT/REQUEST/DATETIME	(#PCDATA) The date and time of the API request. This element appears only when the API request includes the parameter **echo_request=1**.
/VIRTUAL_HOST_LIST_OUTPUT/REQUEST/USER_LOGIN	(#PCDATA) The user login ID of the user who made the request. This element appears only when the API request includes the parameter **echo_request=1**.
/VIRTUAL_HOST_LIST_OUTPUT/REQUEST/RESOURCE	(#PCDATA) The resource specified for the request. This element appears only when the API request includes the parameter **echo_request=**1.
/VIRTUAL_HOST_LIST_OUTPUT/REQUEST/PARAM_LIST	(PARAM+))
/VIRTUAL_HOST_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM	(KEY, VALUE))
/VIRTUAL_HOST_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM/KEY	(#PCDATA) An input parameter name. This element appears only when the API request includes the parameter **echo_request=1**.
/VIRTUAL_HOST_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM/VALUE	(#PCDATA) An input parameter value. This element appears only when the API request includes the parameter **echo_request=1**.
/VIRTUAL_HOST_LIST_OUTPUT/REQUEST/POST_DATA	(#PCDATA) The POST data, if any. This element appears only when the API request includes the parameter **echo_request=1**.
/VIRTUAL_HOST_LIST_OUTPUT/RESPONSE	(DATETIME, (VIRTUAL_HOST_LIST)?, WARNING?)
/VIRTUAL_HOST_LIST_OUTPUT/RESPONSE/DATETIME	(#PCDATA) The date and time of the Qualys response.
/VIRTUAL_HOST_LIST_OUTPUT/RESPONSE/VIRTUAL_HOST_LIST	(VIRTUAL_HOST+)
/VIRTUAL_HOST_LIST_OUTPUT/RESPONSE/VIRTUAL_HOST_LIST/VIRTUAL_HOST	(IP, PORT, FQDN+)
/VIRTUAL_HOST_LIST_OUTPUT/RESPONSE/VIRTUAL_HOST_LIST/VIRTUAL_HOST/IP	(#PCDATA) The IP address for the virtual host configuration.
/VIRTUAL_HOST_LIST_OUTPUT/RESPONSE/VIRTUAL_HOST_LIST/VIRTUAL_HOST/PORT	(#PCDATA) The port for the virtual host configuration.
/VIRTUAL_HOST_LIST_OUTPUT/RESPONSE/VIRTUAL_HOST_LIST/VIRTUAL_HOST/FQDN	(#PCDATA) One FQDN for the virtual host configuration.

IPv6 Mapping Records List Output

The IPv6 mapping records list output identifies IPv6 mapping records in the subscription.

The DTD can be found at the following URL:

```
https://qualysapi.qualys.com/api/2.0/fo/asset/ip/v4_v6/
ip_map_list_output.dtd
```

where <qualysapi.qualys.com> is the API server URL where your account is located.

DTD for IPv6 Mapping Records List Output

A recent DTD for the IPv6 mapping records list output is shown below.

```
<!-- QUALYS IP_MAP_LIST_OUTPUT DTD -->

<!ELEMENT IP_MAP_LIST_OUTPUT (REQUEST?,RESPONSE)>

<!ELEMENT REQUEST (DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?,
                   POST_DATA?)>
<!ELEMENT DATETIME (#PCDATA)>
<!ELEMENT USER_LOGIN (#PCDATA)>
<!ELEMENT RESOURCE (#PCDATA)>
<!ELEMENT PARAM_LIST (PARAM+)>
<!ELEMENT PARAM (KEY, VALUE)>
<!ELEMENT KEY (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>
<!-- if returned, POST_DATA will be urlencoded -->
<!ELEMENT POST_DATA (#PCDATA)>

<!ELEMENT RESPONSE (DATETIME, IP_MAP_LIST?)>

<!ELEMENT IP_MAP_LIST (IP_MAP+)>
<!ELEMENT IP_MAP (ID, V4, V6, NETWORK_ID?)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT V4 (#PCDATA)>
<!ELEMENT V6 (#PCDATA)>

<!-- EOF -->
```

XPaths for IPv6 Mapping Records List Output

This section describes the XPaths for the IPv6 mapping records list output.

XPath	element specifications / notes
/IP_MAP_LIST_OUTPUT	(REQUEST?,RESPONSE)
/IP_MAP_LIST_OUTPUT/REQUEST	(DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?, POST_DATA?)
/IP_MAP_LIST_OUTPUT/REQUEST/DATETIME	(#PCDATA) The date and time of the API request. This element appears only when the API request includes the parameter **echo_request=1**.
/IP_MAP_LIST_OUTPUT/REQUEST/USER_LOGIN	(#PCDATA) The user login ID of the user who made the request. This element appears only when the API request includes the parameter **echo_request=1**.
/IP_MAP_LIST_OUTPUT/REQUEST/RESOURCE	(#PCDATA) The resource specified for the request. This element appears only when the API request includes the parameter **echo_request=**1.
/IP_MAP_LIST_OUTPUT/REQUEST/PARAM_LIST	(PARAM+))
/IP_MAP_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM	(KEY, VALUE))
/IP_MAP_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM/KEY	(#PCDATA) An input parameter name. This element appears only when the API request includes the parameter **echo_request=1**.
/IP_MAP_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM/VALUE	(#PCDATA) An input parameter value. This element appears only when the API request includes the parameter **echo_request=1**.
/IP_MAP_LIST_OUTPUT/REQUEST/POST_DATA	(#PCDATA) The POST data, if any. This element appears only when the API request includes the parameter **echo_request=1**.
/IP_MAP_LIST_OUTPUT/RESPONSE	(DATETIME, IP_MAP_LIST?)
/IP_MAP_LIST_OUTPUT/RESPONSE/DATETIME	(#PCDATA) The date and time of the Qualys response.
/IP_MAP_LIST_OUTPUT/RESPONSE/IP_MAP_LIST_LIST	(IP_MAP+)
/IP_MAP_LIST_OUTPUT/RESPONSE/IP_MAP_LIST_LIST/IP_MAP	(ID, V4, V6)
/IP_MAP_LIST_OUTPUT/RESPONSE/IP_MAP_LIST_LIST/IP_MAP/ID	(#PCDATA) A service-assigned ID for a mapping record.
/IP_MAP_LIST_OUTPUT/RESPONSE/IP_MAP_LIST_LIST/IP_MAP/V4	(#PCDATA) An IPv4 address for a mapping record.
/IP_MAP_LIST_OUTPUT/RESPONSE/IP_MAP_LIST_LIST/IP_MAP/V6	(#PCDATA) An IPv6 address for a mapping record.

Restricted IPs List Output

The DTD for the restricted IPs list output can be found at the following URL:

```
https://qualysapi.qualys.com/api/2.0/fo/setup/restricted_ips/restricted_ips_output.dtd
```

where <qualysapi.qualys.com> is the API server URL where your account is located.

DTD for Restricted IPs List Output

A recent DTD for the restricted IPs list output is shown below.

```
<!-- QUALYS RESTRICTED_IPS_OUTPUT DTD -->

<!ELEMENT RESTRICTED_IPS_OUTPUT (REQUEST?,RESPONSE)>

<!ELEMENT REQUEST (DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?,
                   POST_DATA?)>
<!ELEMENT DATETIME (#PCDATA)>
<!ELEMENT USER_LOGIN (#PCDATA)>
<!ELEMENT RESOURCE (#PCDATA)>
<!ELEMENT PARAM_LIST (PARAM+)>
<!ELEMENT PARAM (KEY, VALUE)>
<!ELEMENT KEY (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>
<!-- if returned, POST_DATA will be urlencoded -->
<!ELEMENT POST_DATA (#PCDATA)>

<!ELEMENT RESPONSE (DATETIME, IP_SET?, STATUS?)>

<!ELEMENT IP_SET ((IP|IP_RANGE)+)>
<!ELEMENT IP (#PCDATA)>
<!ELEMENT IP_RANGE (#PCDATA)>
<!ELEMENT STATUS (#PCDATA)>
<!-- EOF -->
```

XPaths for Restricted IPs List Output

This section describes the XPaths for the restricted IPs list output.

XPath	element specifications / notes
/RESTRICTED_IPS_OUTPUT	(REQUEST?,RESPONSE)
/RESTRICTED_IPS_OUTPUT/REQUEST	(DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?, POST_DATA?)
/RESTRICTED_IPS_OUTPUT/REQUEST/DATETIME	(#PCDATA) The date and time of the API request to download the restricted IPs list. This element appears only when the API request includes the parameter **echo_request=1**.
/RESTRICTED_IPS_OUTPUT/REQUEST/USER_LOGIN	(#PCDATA) The user login ID of the user who made the request. This element appears only when the API request includes the parameter **echo_request=1**.
/RESTRICTED_IPS_OUTPUT/REQUEST/RESOURCE	(#PCDATA) The resource specified for the request. This element appears only when the API request includes the parameter **echo_request=**1.
/RESTRICTED_IPS_OUTPUT/REQUEST/PARAM_LIST	(PARAM+))
/RESTRICTED_IPS_OUTPUT/REQUEST/PARAM_LIST/PARAM	(KEY, VALUE))
/RESTRICTED_IPS_OUTPUT/REQUEST/PARAM_LIST/PARAM/KEY	(#PCDATA) An input parameter name. This element appears only when the API request includes the parameter **echo_request=1**.
/RESTRICTED_IPS_OUTPUT/REQUEST/PARAM_LIST/PARAM/VALUE	(#PCDATA) An input parameter value. This element appears only when the API request includes the parameter **echo_request=1**.
/RESTRICTED_IPS_OUTPUT/REQUEST/POST_DATA	(#PCDATA) The POST data, if any. This element appears only when the API request includes the parameter **echo_request=1**.
/RESTRICTED_IPS_OUTPUT/RESPONSE	(DATETIME, IP_SET?, STATUS?)
/RESTRICTED_IPS_OUTPUT/RESPONSE/DATETIME	(#PCDATA) The date and time of the Qualys response.
/RESTRICTED_IPS_OUTPUT/RESPONSE/IP_SET	((IP|IP_RANGE)+)
/RESTRICTED_IPS_OUTPUT/RESPONSE/IP_SET/IP	(#PCDATA) An IP address in the restricted IPs list.
/RESTRICTED_IPS_OUTPUT/RESPONSE/IP_SET/IP_RANGE	(#PCDATA) An IP address range in the restricted IPs list.
/RESTRICTED_IPS_OUTPUT/RESPONSE/STATUS	(#PCDATA) The status of the restricted IPs list: enabled or disabled. When enabled a user who attempts to log in to Qualys from an IP in the restricted IPs list will be denied access.

Asset Group List Output

The DTD for the asset group list output can be found at the following URL (where <qualysapi.qualys.com> is the API server URL where your account is located):

```
https://qualysapi.qualys.com/api/2.0/fo/asset/group/asset_group_list_output.dtd
```

DTD for Asset Group List Output

A recent DTD for the asset group list output is below.

```
<!-- QUALYS ASSET_GROUP_LIST_OUTPUT DTD -->

<!ELEMENT ASSET_GROUP_LIST_OUTPUT (REQUEST?,RESPONSE)>

<!ELEMENT REQUEST (DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?,
                   POST_DATA?)>
<!ELEMENT DATETIME (#PCDATA)>
<!ELEMENT USER_LOGIN (#PCDATA)>
<!ELEMENT RESOURCE (#PCDATA)>
<!ELEMENT PARAM_LIST (PARAM+)>
<!ELEMENT PARAM (KEY, VALUE)>
<!ELEMENT KEY (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>
<!-- if returned, POST_DATA will be urlencoded -->
<!ELEMENT POST_DATA (#PCDATA)>

<!ELEMENT RESPONSE (DATETIME, (ASSET_GROUP_LIST|ID_SET)?, WARNING?)>
<!ELEMENT ASSET_GROUP_LIST (ASSET_GROUP+)>
<!ELEMENT ASSET_GROUP (ID, TITLE?, OWNER_ID?, UNIT_ID?,
                (NETWORK_ID|NETWORK_IDS)?, LAST_UPDATE?,
                BUSINESS_IMPACT?, CVSS_ENVIRO_CDP?, CVSS_ENVIRO_TD?,
                CVSS_ENVIRO_CR?, CVSS_ENVIRO_IR?, CVSS_ENVIRO_AR?,
                DEFAULT_APPLIANCE_ID?, APPLIANCE_IDS?,
                IP_SET?, DOMAIN_LIST?, DNS_LIST?, NETBIOS_LIST?,
                EC2_ID_LIST?, HOST_IDS?,
                USER_IDS?, UNIT_IDS?)>

<!ELEMENT ID (#PCDATA)>
<!ELEMENT TITLE (#PCDATA)>
<!ELEMENT OWNER_ID (#PCDATA)>
<!ELEMENT UNIT_ID (#PCDATA)>
<!ELEMENT NETWORK_ID (#PCDATA)>
<!ELEMENT NETWORK_IDS (#PCDATA)>
<!ELEMENT LAST_UPDATE (#PCDATA)>
<!ELEMENT BUSINESS_IMPACT (#PCDATA)>
```

```
<!-- CVSS -->
<!ELEMENT CVSS_ENVIRO_CDP (#PCDATA)>
<!ELEMENT CVSS_ENVIRO_TD (#PCDATA)>
<!ELEMENT CVSS_ENVIRO_CR (#PCDATA)>
<!ELEMENT CVSS_ENVIRO_IR (#PCDATA)>
<!ELEMENT CVSS_ENVIRO_AR (#PCDATA)>

<!-- APPLIANCE_LIST -->
<!ELEMENT DEFAULT_APPLIANCE_ID (#PCDATA)>
<!ELEMENT APPLIANCE_IDS (#PCDATA)>

<!-- IP_SET -->
<!ELEMENT IP_SET ((IP|IP_RANGE)+)>
<!ELEMENT IP (#PCDATA)>
<!ATTLIST IP network_id CDATA "0">
<!ELEMENT IP_RANGE (#PCDATA)>
<!ATTLIST IP_RANGE network_id CDATA "0">

<!-- DOMAIN_LIST -->
<!ELEMENT DOMAIN_LIST (DOMAIN+)>
<!ELEMENT DOMAIN (#PCDATA)>
<!ATTLIST DOMAIN netblock CDATA "">
<!ATTLIST DOMAIN network_id CDATA "0">

<!-- DNS_LIST -->
<!ELEMENT DNS_LIST (DNS+)>
<!ELEMENT DNS (#PCDATA)>
<!ATTLIST DNS network_id CDATA "0">

<!-- NETBIOS_LIST -->
<!ELEMENT NETBIOS_LIST (NETBIOS+)>
<!ELEMENT NETBIOS (#PCDATA)>
<!ATTLIST NETBIOS network_id CDATA "0">

<!-- EC2_IDS -->
<!ELEMENT EC2_IDS (#PCDATA)>

<!-- HOST_IDS -->
<!ELEMENT HOST_IDS (#PCDATA)>

<!-- USER_IDS -->
<!ELEMENT USER_IDS (#PCDATA)>

<!-- UNIT_IDS -->
<!ELEMENT UNIT_IDS (#PCDATA)>

<!-- WARNING -->
<!ELEMENT WARNING (CODE?, TEXT, URL?)>
```

```
<!ELEMENT CODE (#PCDATA)>
<!ELEMENT TEXT (#PCDATA)>
<!ELEMENT URL (#PCDATA)>
```

XPaths for Asset Group List Output

This section describes the XPaths for the asset group list output.

XPath	element specifications / notes
/ASSET_GROUP_LIST_OUTPUT	(REQUEST?,RESPONSE)
/ASSET_GROUP_LIST_OUTPUT/REQUEST	(DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?, POST_DATA?)
/ASSET_GROUP_LIST_OUTPUT/REQUEST/DATETIME	(#PCDATA) The date and time of the API reques. This element appears only when the API request includes the parameter **echo_request=1**.
/ASSET_GROUP_LIST_OUTPUT/REQUEST/USER_LOGIN	(#PCDATA) The user login ID of the user who made the request. This element appears only when the API request includes the parameter **echo_request=1**.
/ASSET_GROUP_LIST_OUTPUT/REQUEST/RESOURCE	(#PCDATA) The resource specified for the request. This element appears only when the API request includes the parameter **echo_request=**1.
/ASSET_GROUP_LIST_OUTPUT/REQUEST/PARAM_LIST	(PARAM+))
/ASSET_GROUP_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM	(KEY, VALUE))
/ASSET_GROUP_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM/KEY	(#PCDATA) An input parameter name. This element appears only when the API request includes the parameter **echo_request=1**.
/ASSET_GROUP_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM/VALUE	(#PCDATA) An input parameter value. This element appears only when the API request includes the parameter **echo_request=1**.
/ASSET_GROUP_LIST_OUTPUT/REQUEST/POST_DATA	(#PCDATA) The POST data, if any. This element appears only when the API request includes the parameter **echo_request=1**.
/ASSET_GROUP_LIST_OUTPUT/RESPONSE	(DATETIME, (ASSET_GROUP_LIST|ID_SET)?, WARNING?)
/ASSET_GROUP_LIST_OUTPUT/RESPONSE/DATETIME	(#PCDATA) The date and time of the Qualys response.
/ASSET_GROUP_LIST_OUTPUT/RESPONSE/ASSET_GROUP_LIST	(ASSET_GROUP+)
/ASSET_GROUP_LIST_OUTPUT/RESPONSE/ASSET_GROUP_LIST/ASSET_GROUP	(ID, TITLE?, OWNER_ID?, UNIT_ID?, (NETWORK_ID|NETWORK_IDS)?, LAST_UPDATE?, BUSINESS_IMPACT?, CVSS_ENVIRO_CDP?, CVSS_ENVIRO_TD?, CVSS_ENVIRO_CR?, CVSS_ENVIRO_IR?, CVSS_ENVIRO_AR?, DEFAULT_APPLIANCE_ID?, APPLIANCE_IDS?, IP_SET?, DOMAIN_LIST?, DNS_LIST?, NETBIOS_LIST?, EC2_ID_LIST?, HOST_IDS?, USER_IDS?, UNIT_IDS?)

XPath	element specifications / notes
/ASSET_GROUP_LIST_OUTPUT/RESPONSE/ASSET_GROUP_LIST/ASSET_GROUP/ID (#PCDATA)	
	The ID of the asset group.
/ASSET_GROUP_LIST_OUTPUT/RESPONSE/ASSET_GROUP_LIST/ASSET_GROUP/TITLE (#PCDATA)	
	The title of the asset group.
/ASSET_GROUP_LIST_OUTPUT/RESPONSE/ASSET_GROUP_LIST/ASSET_GROUP/OWNER_ID (#PCDATA)	
	The ID of the asset group's owner.
/ASSET_GROUP_LIST_OUTPUT/RESPONSE/ASSET_GROUP_LIST/ASSET_GROUP/UNIT_ID (#PCDATA)	
	The business unit ID of the asset group.
/ASSET_GROUP_LIST_OUTPUT/RESPONSE/ASSET_GROUP_LIST/ASSET_GROUP/NETWORK_ID (#PCDATA)	
	(Appears only if the Networks feature is enabled for your subscription) The asset group will be assigned to a custom network ID or 0 (the Global Default Network).
/ASSET_GROUP_LIST_OUTPUT/RESPONSE/ASSET_GROUP_LIST/ASSET_GROUP/NETWORK_IDS (#PCDATA)	
	(Appears only if the Networks feature is enabled for your subscription) This element lists custom network IDs that include the All asset group. Have multiple All asset groups? Yes you might. There is 1 All group for the subscription, and 1 All group for each custom business unit.
/ASSET_GROUP_LIST_OUTPUT/RESPONSE/ASSET_GROUP_LIST/ASSET_GROUP/ BUSINESS_IMPACT (#PCDATA)	
	The business impact assigned to the asset group.
/ASSET_GROUP_LIST_OUTPUT/RESPONSE/ASSET_GROUP_LIST/ASSET_GROUP/CVSS<value> (#PCDATA)	
	The CVSS environmental metrics assigned to the asset group. CVSS_ENVIRO_CDP (Collateral Damage Potential) CVSS_ENVIRO_TD (Target Distribution) CVSS_ENVIRO_CR (Confidentiality Requirement) CVSS_ENVIRO_IR (Integrity Requirement) CVSS_ENVIRO_AR (Availability Requirement)
/ASSET_GROUP_LIST_OUTPUT/RESPONSE/ASSET_GROUP_LIST/ASSET_GROUP/ DEFAULT_APPLIANCE_ID (#PCDATA)	
	The ID of the asset group's default scanner appliance.
/ASSET_GROUP_LIST_OUTPUT/RESPONSE/ASSET_GROUP_LIST/ASSET_GROUP/ APPLIANCE_IDS (#PCDATA)	
	The IDs of the scanner appliances assigned to the asset group.
/ASSET_GROUP_LIST_OUTPUT/RESPONSE/ASSET_GROUP_LIST/ASSET_GROUP/IP_SET (IP|IP_RANGE)	
/ASSET_GROUP_LIST_OUTPUT/RESPONSE/ASSET_GROUP_LIST/ASSET_GROUP/IP_SET/IP (#PCDATA)	
	An IP address assigned to the asset group. If the Networks feature is enabled in your subscription, the attribute "network_id" is the network ID associated with this IP address.
/ASSET_GROUP_LIST_OUTPUT/RESPONSE/ASSET_GROUP_LIST/ASSET_GROUP/ IP_SET/IP_RANGE (#PCDATA)	
	An IP address range assigned to the asset group. If the Networks feature is enabled in your subscription, the attribute "network_id" is the network ID associated with this IP range.

XPath	element specifications / notes
/ASSET_GROUP_LIST_OUTPUT/RESPONSE/ASSET_GROUP_LIST/ASSET_GROUP/DOMAIN_LIST (DOMAIN+)	
/ASSET_GROUP_LIST_OUTPUT/RESPONSE/ASSET_GROUP_LIST/ASSET_GROUP/DOMAIN_LIST/DOMAIN (#PCDATA)	
	A domain assigned to the asset group. The attribute "netblock" is the netblock assigned to this domain, if any. If the Networks feature is enabled in your subscription, the attribute "network_id" is the network ID associated with this IP address.
/ASSET_GROUP_LIST_OUTPUT/RESPONSE/ASSET_GROUP_LIST/ASSET_GROUP/DNS_LIST (DNS+)	
/ASSET_GROUP_LIST_OUTPUT/RESPONSE/ASSET_GROUP_LIST/ASSET_GROUP/DNS_LIST/DNS (#PCDATA)	
	A DNS name assigned to the asset group. If the Networks feature is enabled in your subscription, the attribute "network_id" is the network ID associated with the DNS host.
/ASSET_GROUP_LIST_OUTPUT/RESPONSE/ASSET_GROUP_LIST/ASSET_GROUP/NETBIOS_LIST (NETBIOS+)	
/ASSET_GROUP_LIST_OUTPUT/RESPONSE/ASSET_GROUP_LIST/ASSET_GROUP/NETBIOS_LIST/NETBIOS (#PCDATA)	
	A NetBIOS name assigned to the asset group. If the Networks feature is enabled in your subscription, the attribute "network_id" is the network ID associated with the NetBIOS host.
/ASSET_GROUP_LIST_OUTPUT/RESPONSE/ASSET_GROUP_LIST/ASSET_GROUP/EC2_IDS (#PCDATA)	
	EC2 IDs associated with the asset group.
/ASSET_GROUP_LIST_OUTPUT/RESPONSE/ASSET_GROUP_LIST/ASSET_GROUP/EC2_IDS (#PCDATA)	
	The asset group is associated with these EC2 IDs.
/ASSET_GROUP_LIST_OUTPUT/RESPONSE/ASSET_GROUP_LIST/ASSET_GROUP/HOST_IDS (#PCDATA)	
	The host IDs associated with the asset group.
/ASSET_GROUP_LIST_OUTPUT/RESPONSE/ASSET_GROUP_LIST/ASSET_GROUP/USER_IDS (#PCDATA)	
	The asset group is visible to users with these user IDs.
/ASSET_GROUP_LIST_OUTPUT/RESPONSE/ASSET_GROUP_LIST/ASSET_GROUP/UNIT_IDS (#PCDATA)	
	The asset group is assigned to business units with these unit IDs.
/ASSET_GROUP_LIST_OUTPUT/RESPONSE/WARNING (CODE?, TEXT, URL?)	
/ASSET_GROUP_LIST_OUTPUT/RESPONSE/WARNING/CODE (#PCDATA)	
	The warning code. This code appears when the API request finds more than 1,000 asset group records.
/ASSET_GROUP_LIST_OUTPUT/RESPONSE/WARNING/TEXT (#PCDATA)	
	The warning message text. This message appears when the API request finds more than 1,000 asset group records.
/ASSET_GROUP_LIST_OUTPUT/RESPONSE/WARNING/URL (#PCDATA)	
	The URL for making another request for the next batch of asset group records.

APPENDIX C

Compliance Data XML

This appendix describes the XML output returned from API V2 requests for compliance data using the Policy Compliance API functions.

- Compliance Control List Output
- Compliance Policy List Output
- Compliance Policy Export Output
- Compliance Posture Information Output
- Compliance Policy Report
- Compliance Authentication Report
- Compliance Scorecard Report
- SCAP Policy List Output

Compliance Control List Output

The compliance control list XML is returned from a compliance control list API call. The DTD can be found at the following URL (where qualysapi.qualys.com is the API server where your account is located):

```
https://qualysapi.qualys.com/api/2.0/fo/compliance/control/control_list_output.dtd
```

DTD for Compliance Control List Output

A recent DTD for the compliance control list is shown below.

```
<!-- QUALYS CONTROL_LIST_OUTPUT DTD -->

<!ELEMENT CONTROL_LIST_OUTPUT (REQUEST?,RESPONSE)>

<!ELEMENT REQUEST (DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?,
                   POST_DATA?)>
<!ELEMENT DATETIME (#PCDATA)>
<!ELEMENT USER_LOGIN (#PCDATA)>
<!ELEMENT RESOURCE (#PCDATA)>
<!ELEMENT PARAM_LIST (PARAM+)>
<!ELEMENT PARAM (KEY, VALUE)>
<!ELEMENT KEY (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>
<!-- if returned, POST_DATA will be urlencoded -->
<!ELEMENT POST_DATA (#PCDATA)>

<!ELEMENT RESPONSE (DATETIME, (CONTROL_LIST|ID_SET)?, WARNING?)>
<!ELEMENT CONTROL_LIST (CONTROL+)>
<!ELEMENT CONTROL (ID, UPDATE_DATE, CREATED_DATE, CATEGORY, SUB_CATEGORY,
                   STATEMENT, CRITICALITY?, DEPRECATED?,
                   DEPRECATED_DATE?, TECHNOLOGY_LIST, FRAMEWORK_LIST?)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT UPDATE_DATE (#PCDATA)>
<!ELEMENT CREATED_DATE (#PCDATA)>
<!ELEMENT CATEGORY (#PCDATA)>
<!ELEMENT SUB_CATEGORY (#PCDATA)>
<!ELEMENT STATEMENT (#PCDATA)>
<!ELEMENT CRITICALITY (LABEL, VALUE)>
<!ELEMENT LABEL (#PCDATA)>
<!ELEMENT DEPRECATED (#PCDATA)>
<!ELEMENT DEPRECATED_DATE (#PCDATA)>
<!ELEMENT TECHNOLOGY_LIST (TECHNOLOGY+)>
<!ELEMENT TECHNOLOGY (ID, NAME, RATIONALE)>
<!ELEMENT NAME (#PCDATA)>
```

```
<!ELEMENT RATIONALE (#PCDATA)>
<!ELEMENT FRAMEWORK_LIST (FRAMEWORK+)>
<!ELEMENT FRAMEWORK (ID, NAME, REFERENCE_LIST)>
<!ELEMENT REFERENCE_LIST (REFERENCE+)>
<!ELEMENT REFERENCE (SECTION, COMMENTS)>
<!ELEMENT SECTION (#PCDATA)>
<!ELEMENT COMMENTS (#PCDATA)>

<!ELEMENT ID_SET (ID|ID_RANGE)+>
<!ELEMENT ID_RANGE (#PCDATA)>

<!ELEMENT WARNING (CODE?, TEXT, URL?)>
<!ELEMENT CODE (#PCDATA)>
<!ELEMENT TEXT (#PCDATA)>
<!ELEMENT URL (#PCDATA)>
<!-- EOF -->
```

XPaths for Compliance Control List Output

This section describes the XPaths for the compliance control list output.

Compliance Control List Output: Request

XPath	element specifications / notes
/COMPLIANCE_CONTROL_LIST_OUTPUT	(REQUEST?, RESPONSE)
/COMPLIANCE_CONTROL_LIST_OUTPUT/REQUEST	(DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?, POST_DATA?)
/COMPLIANCE_CONTROL_LIST_OUTPUT/REQUEST/DATETIME	(#PCDATA) The date and time of the request.
/COMPLIANCE_CONTROL_LIST_OUTPUT/REQUEST/USER_LOGIN	(#PCDATA) The user login ID of the user who made the request.
/COMPLIANCE_CONTROL_LIST_OUTPUT/REQUEST/RESOURCE	(#PCDATA) The resource specified for the request.
/COMPLIANCE_CONTROL_LIST_OUTPUT/REQUEST/PARAM_LIST	(PARAM+)
/COMPLIANCE_CONTROL_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM	(KEY, VALUE)
/COMPLIANCE_CONTROL_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM/KEY	(#PCDATA) An input parameter name.
/COMPLIANCE_CONTROL_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM/VALUE	(#PCDATA) An input parameter value.
/COMPLIANCE_CONTROL_LIST_OUTPUT/REQUEST/POST_DATA	(#PCDATA) The POST data, if any.

Compliance Control List Output: Response

XPath	element specifications / notes
/COMPLIANCE_CONTROL_LIST_OUTPUT (REQUEST?, RESPONSE)	
/COMPLIANCE_CONTROL_LIST_OUTPUT/RESPONSE	(DATETIME, CONTROL_LIST|ID_SET?, WARNING?)
/COMPLIANCE_CONTROL_LIST_OUTPUT/RESPONSE/DATETIME (#PCDATA)	The date and time of the response.
/COMPLIANCE_CONTROL_LIST_OUTPUT/RESPONSE/CONTROL_LIST (CONTROL+)	
/COMPLIANCE_CONTROL_LIST_OUTPUT/RESPONSE/CONTROL_LIST/CONTROL	(ID, UPDATE_DATE, CREATED_DATE, CATEGORY, SUB_CATEGORY, STATEMENT, CRITICALITY?, DEPRECATED?, DEPRECATED_DATE?, TECHNOLOGY_LIST, FRAMEWORK_LIST?)
/COMPLIANCE_CONTROL_LIST_OUTPUT/RESPONSE/CONTROL_LIST/CONTROL/ID (#PCDATA)	A compliance control ID.
/COMPLIANCE_CONTROL_LIST_OUTPUT/RESPONSE/CONTROL_LIST/CONTROL/UPDATE_DATE (#PCDATA)	The date and time when the control was last updated.
/COMPLIANCE_CONTROL_LIST_OUTPUT/RESPONSE/CONTROL_LIST/CONTROL/CREATED_DATE (#PCDATA)	The date and time when the control was created.
/COMPLIANCE_CONTROL_LIST_OUTPUT/RESPONSE/CONTROL_LIST/CONTROL/CATEGORY (#PCDATA)	A category for a compliance control.
/COMPLIANCE_CONTROL_LIST_OUTPUT/RESPONSE/CONTROL_LIST/CONTROL/SUB-CATEGORY	(#PCDATA) A sub-category for a compliance control.
/COMPLIANCE_CONTROL_LIST_OUTPUT/RESPONSE/CONTROL_LIST/CONTROL/STATEMENT (#PCDATA)	A statement for a compliance control.
/COMPLIANCE_CONTROL_LIST_OUTPUT/RESPONSE/CONTROL_LIST/CONTROL/CRITICALITY	(LABEL, VALUE)
/COMPLIANCE_CONTROL_LIST_OUTPUT/RESPONSE/CONTROL_LIST/CONTROL/CRITICALITY/LABEL (#PCDATA)	A criticality label (e.g. SERIOUS, CRITICAL, URGENT) assigned to the control.
/COMPLIANCE_CONTROL_LIST_OUTPUT/RESPONSE/CONTROL_LIST/CONTROL/CRITICALITY/VALUE (#PCDATA)	A criticality value (0-5) assigned to the control.
/COMPLIANCE_CONTROL_LIST_OUTPUT/RESPONSE/CONTROL_LIST/CONTROL/ DEPRECATED (#PCDATA)	The value 1 identifies a deprecated control. This element appears only for a deprecated control.

XPath	element specifications / notes
/COMPLIANCE_CONTROL_LIST_OUTPUT/RESPONSE/CONTROL_LIST/CONTROL/DEPRECATED_DATE (#PCDATA)	
	For a deprecated control, the date the control was deprecated. This element appears only for a deprecated control
/COMPLIANCE_CONTROL_LIST_OUTPUT/RESPONSE/CONTROL_LIST/CONTROL/TECHNOLOGY_LIST	(TECHNOLOGY+)
/COMPLIANCE_CONTROL_LIST_OUTPUT/RESPONSE/CONTROL_LIST/CONTROL/TECHNOLOGY_LIST/TECHNOLOGY	(ID, NAME, RATIONALE)
/COMPLIANCE_CONTROL_LIST_OUTPUT/RESPONSE/CONTROL_LIST/CONTROL/TECHNOLOGY_LIST/TECHNOLOGY/ID	(#PCDATA)
	A technology ID for a technology in a control.
/COMPLIANCE_CONTROL_LIST_OUTPUT/RESPONSE/CONTROL_LIST/CONTROL/TECHNOLOGY_LIST/TECHNOLOGY/NAME	(#PCDATA)
	A technology name for a technology in a control.
/COMPLIANCE_CONTROL_LIST_OUTPUT/RESPONSE/CONTROL_LIST/CONTROL/TECHNOLOGY_LIST/TECHNOLOGY/RATIONALE (#PCDATA)	
	The rationale description for a technology in a control.
/COMPLIANCE_CONTROL_LIST_OUTPUT/RESPONSE/CONTROL_LIST/CONTROL/FRAMEWORK_LIST	(FRAMEWORK+)
/COMPLIANCE_CONTROL_LIST_OUTPUT/RESPONSE/CONTROL_LIST/CONTROL/FRAMEWORK_LIST/FRAMEWORK (ID, NAME, REFERENCE_LIST)	
/COMPLIANCE_CONTROL_LIST_OUTPUT/RESPONSE/CONTROL_LIST/CONTROL/FRAMEWORK_LIST/FRAMEWORK/ID (#PCDATA)	
	A framework ID for a framework reference in a control.
/COMPLIANCE_CONTROL_LIST_OUTPUT/RESPONSE/CONTROL_LIST/CONTROL/FRAMEWORK_LIST/FRAMEWORK/NAME (#PCDATA)	
	A framework name for a framework reference in a control.
/COMPLIANCE_CONTROL_LIST_OUTPUT/RESPONSE/CONTROL_LIST/CONTROL/FRAMEWORK_LIST/FRAMEWORK/REFERENCE_LIST (REFERENCE+)	
/COMPLIANCE_CONTROL_LIST_OUTPUT/RESPONSE/CONTROL_LIST/CONTROL/FRAMEWORK_LIST/FRAMEWORK/REFERENCE_LIST/REFERENCE (SECTION, COMMENTS)	
/COMPLIANCE_CONTROL_LIST_OUTPUT/RESPONSE/CONTROL_LIST/CONTROL/FRAMEWORK_LIST/FRAMEWORK/REFERENCE_LIST/REFERENCE/SECTION (#PCDATA)	
	A framework section for a framework reference in a control.
/COMPLIANCE_CONTROL_LIST_OUTPUT/RESPONSE/CONTROL_LIST/CONTROL/FRAMEWORK_LIST/FRAMEWORK/REFERENCE_LIST/REFERENCE/COMMENTS (#PCDATA)	
	A framework description (comments) for a framework reference in a control.
/COMPLIANCE_CONTROL_LIST_OUTPUT/RESPONSE/ID_SET (ID|ID_RANGE)+	
/COMPLIANCE_CONTROL_LIST_OUTPUT/RESPONSE/ID_SET/ID (#PCDATA)	
	A compliance control ID.

XPath	element specifications / notes
/COMPLIANCE_CONTROL_LIST_OUTPUT/RESPONSE/ID_SET/ID_RANGE	(#PCDATA)
	A range of compliance control IDs.

Compliance Control List Output: Warning

XPath	element specifications / notes
/COMPLIANCE_CONTROL_LIST_OUTPUT/RESPONSE/WARNING	(CODE, TEXT, URL?)
/COMPLIANCE_CONTROL_LIST_OUTPUT/RESPONSE/WARNING/CODE	(#PCDATA)
	A warning code. A warning code appears when the API request identifies more than 1,000 records (controls).
/COMPLIANCE_CONTROL_LIST_OUTPUT/RESPONSE/WARNING/TEXT	(#PCDATA)
	A warning message. A warning message appears when the API request identifies more than 1,000 records (controls).
/COMPLIANCE_CONTROL_LIST_OUTPUT/RESPONSE/WARNING/URL	(#PCDATA)
	The URL for making another API request for the next batch of compliance control records.

Compliance Policy List Output

The compliance policy list XML is returned from an API request for a compliance policy list. The DTD can be found at the following URL (where qualysapi.qualys.com is the API server URL where your account is located):

```
https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/policy_list_output.dtd
```

DTD for Compliance Policy List Output

A recent DTD for the compliance policy list is shown below.

```
<!-- QUALYS POLICY_LIST_OUTPUT DTD -->

<!ELEMENT POLICY_LIST_OUTPUT (REQUEST?,RESPONSE)>

<!ELEMENT REQUEST (DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?,
                   POST_DATA?)>
<!ELEMENT DATETIME (#PCDATA)>
<!ELEMENT USER_LOGIN (#PCDATA)>
<!ELEMENT RESOURCE (#PCDATA)>
<!ELEMENT PARAM_LIST (PARAM+)>
<!ELEMENT PARAM (KEY, VALUE)>
<!ELEMENT KEY (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>
<!-- if returned, POST_DATA will be urlencoded -->
<!ELEMENT POST_DATA (#PCDATA)>

<!ELEMENT RESPONSE (DATETIME, (POLICY_LIST|ID_SET)?, WARNING_LIST?,
                    GLOSSARY?)>
<!ELEMENT POLICY_LIST (POLICY+)>
<!ELEMENT POLICY (ID, TITLE, CREATED?, LAST_MODIFIED?, STATUS?,
                  ASSET_GROUP_IDS?, CONTROL_LIST?)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT TITLE (#PCDATA)>

<!ELEMENT CREATED (DATETIME, BY)>
<!ELEMENT BY (#PCDATA)>

<!ELEMENT LAST_MODIFIED (DATETIME, BY)>

<!ELEMENT STATUS (#PCDATA)>

<!ELEMENT ASSET_GROUP_IDS (#PCDATA)>
<!ATTLIST ASSET_GROUP_IDS has_hidden_data CDATA #IMPLIED>
```

```
<!ELEMENT CONTROL_LIST (CONTROL+)>
<!ELEMENT CONTROL (ID, STATEMENT, CRITICALITY?, DEPRECATED?,
                   TECHNOLOGY_LIST?)>
<!ELEMENT STATEMENT (#PCDATA)>
<!ELEMENT CRITICALITY (LABEL, VALUE)>
<!ELEMENT LABEL (#PCDATA)>
<!ELEMENT DEPRECATED (#PCDATA)>

<!ELEMENT TECHNOLOGY_LIST (TECHNOLOGY+)>
<!ELEMENT TECHNOLOGY (ID, NAME, RATIONALE, CUSTOMIZED)>
<!ELEMENT NAME (#PCDATA)>
<!ELEMENT RATIONALE (#PCDATA)>
<!ELEMENT CUSTOMIZED (#PCDATA)>

<!ELEMENT ID_SET (ID|ID_RANGE)+>
<!ELEMENT ID_RANGE (#PCDATA)>

<!ELEMENT WARNING_LIST (WARNING+)>
<!ELEMENT WARNING (CODE?, TEXT, URL?)>
<!ELEMENT CODE (#PCDATA)>
<!ELEMENT TEXT (#PCDATA)>
<!ELEMENT URL (#PCDATA)>

<!ELEMENT GLOSSARY (ASSET_GROUP_LIST?, USER_LIST?)>

<!ELEMENT ASSET_GROUP_LIST (ASSET_GROUP+)>
<!ELEMENT ASSET_GROUP (ID, TITLE, NETWORK_ID?, IP_SET?)>
<!ELEMENT NETWORK_ID (#PCDATA)>
<!ELEMENT IP_SET (IP|IP_RANGE)+>
<!ELEMENT IP (#PCDATA)>
<!ELEMENT IP_RANGE (#PCDATA)>

<!ELEMENT USER_LIST (USER+)>
<!ELEMENT USER (USER_LOGIN, FIRST_NAME, LAST_NAME)>
<!ELEMENT FIRST_NAME (#PCDATA)>
<!ELEMENT LAST_NAME (#PCDATA)>

<!-- EOF -->
```

XPaths for Compliance Policy List Output

This section describes the XPaths for the compliance policy list output.

Compliance Policy List Output: Request

XPath	element specifications / notes
/COMPLIANCE_POLICY_LIST_OUTPUT	(REQUEST?, RESPONSE)
/COMPLIANCE_POLICY_LIST_OUTPUT/REQUEST	(DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?, POST_DATA?)
/COMPLIANCE_POLICY_LIST_OUTPUT/REQUEST/DATETIME	(#PCDATA) The date and time of the request.
/COMPLIANCE_POLICY_LIST_OUTPUT/REQUEST/USER_LOGIN	(#PCDATA) The user login ID of the user who made the request.
/COMPLIANCE_POLICY_LIST_OUTPUT/REQUEST/RESOURCE	(#PCDATA) The resource specified for the request.
/COMPLIANCE_POLICY_LIST_OUTPUT/REQUEST/PARAM_LIST	(PARAM+)
/COMPLIANCE_POLICY_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM	(KEY, VALUE)
/COMPLIANCE_POLICY_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM/KEY	(#PCDATA) An input parameter name.
/COMPLIANCE_POLICY_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM/VALUE	(#PCDATA) An input parameter value.
/COMPLIANCE_POLICY_LIST_OUTPUT/REQUEST/POST_DATA	(#PCDATA) The POST data, if any.

Compliance Policy List Output: Response

XPath	element specifications / notes
/COMPLIANCE_POLICY_LIST_OUTPUT	(REQUEST?, RESPONSE)
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE	(DATETIME, (POLICY_LIST|ID_SET)?, WARNING?, GLOSSARY?)
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/DATETIME	(#PCDATA) The date and time of the response.
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/POLICY_LIST	(POLICY+)
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/POLICY_LIST/POLICY	(ID, TITLE, CREATED?, LAST_MODIFIED?, STATUS?, ASSET_GROUP_IDS?, CONTROL_LIST?)
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/POLICY_LIST/POLICY/ID	(#PCDATA) A compliance policy ID.

XPath	element specifications / notes
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/POLICY_LIST/POLICY/TITLE (#PCDATA)	
	A compliance policy title.
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/POLICY_LIST/POLICY/CREATED (#PCDATA)	
	The date/time when the policy was created.
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/POLICY_LIST/POLICY/LAST_MODIFIED (DATETIME, BY)	
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/POLICY_LIST/POLICY/LAST_MODIFIED/DATETIME (#PCDATA)	
	The date/time when the policy was last updated.
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/POLICY_LIST/POLICY/LAST_MODIFIED/BY (#PCDATA)	
	The user login ID of the user who last modified the policy.
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/POLICY_LIST/POLICY/STATUS (#PCDATA)	
	The current status of the policy: active or inactive.
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/POLICY_LIST/POLICY/ASSET_GROUP_IDS (#PCDATA)	
	A list of asset group IDs for the asset groups assigned to a policy.
attribute: **has_hidden_data**	**has_hidden_data** is *implied* and, if present, has the value 1. This flag indicates that the user does not have permission to see one or more asset groups in the policy. When this attribute is present, only the asset group IDs that the user has permission to see, if any, are listed in the <ASSET_GROUP_IDS> element.
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/POLICY_LIST/POLICY/CONTROL_LIST (CONTROL+)	
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/POLICY_LIST/POLICY/CONTROL_LIST/CONTROL	
	(ID, STATEMENT, CRITICALITY?, DEPRECATED?, TECHNOLOGY_LIST?)
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/POLICY_LIST/POLICY/CONTROL_LIST/CONTROL/ID	
	(#PCDATA)
	A compliance control ID.
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/POLICY_LIST/POLICY/CONTROL_LIST/CONTROL/STATEMENT (#PCDATA)	
	A control statement.
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/POLICY_LIST/POLICY/CONTROL_LIST/CONTROL/CRITICALITY (LABEL, VALUE)	
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/POLICY_LIST/POLICY/CONTROL_LIST/CONTROL/CRITICALITY/LABEL (#PCDATA)	
	A criticality label (e.g. SERIOUS, CRITICAL, URGENT) assigned to the control.
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/POLICY_LIST/POLICY/CONTROL_LIST/CONTROL/CRITICALITY/VALUE (#PCDATA)	
	A criticality value (0-5) assigned to the control.
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/POLICY_LIST/POLICY/CONTROL_LIST/CONTROL/DEPRECATED (#PCDATA)	
	The value 1 identifies a deprecated control. This element appears only for a deprecated control.

XPath	element specifications / notes
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/POLICY_LIST/POLICY/CONTROL_LIST/CONTROL/TECHNOLOGY_LIST	(TECHNOLOGY+)
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/POLICY_LIST/POLICY/CONTROL_LIST/CONTROL/TECHNOLOGY_LIST/TECHNOLOGY	(ID, NAME, RATIONALE, CUSTOMIZED)
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/POLICY_LIST/POLICY/CONTROL_LIST/CONTROL/TECHNOLOGY_LIST/TECHNOLOGY/ID	(#PCDATA)
	A technology ID for a control.
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/POLICY_LIST/POLICY/CONTROL_LIST/CONTROL/TECHNOLOGY_LIST/TECHNOLOGY/NAME	(#PCDATA)
	A technology name for a control.
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/POLICY_LIST/POLICY/CONTROL_LIST/CONTROL/TECHNOLOGY_LIST/TECHNOLOGY/RATIONALE	(#PCDATA)
	The rationale description for a control technology.
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/POLICY_LIST/POLICY/CONTROL_LIST/CONTROL/TECHNOLOGY_LIST/TECHNOLOGY/CUSTOMIZED	(#PCDATA)
	A value indicating whether the default value was customized for a control technology. The value 1 indicates the default value was customized. The value 0 indicates the default value was not customized. The value 0 always is present for a locked control (a control that cannot be customized).
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/ID_SET	(ID|ID_RANGE)
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/ID_SET/ID	(#PCDATA)
	A policy ID.
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/ID_SET/ID_RANGE	(#PCDATA)
	A range policy IDs.

Compliance Policy List Output: Warning

XPath	element specifications / notes
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/WARNING_LIST	(WARNING+)
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/WARNING_LIST/WARNING	(CODE?, TEXT, URL?)
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/WARNING/CODE	(#PCDATA)
	A warning code. A warning code appears when the API request identifies more than 1,000 records (policies).
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/WARNING/TEXT	(#PCDATA)
	A warning message. A warning message appears when the API request identifies more than 1,000 records (policies).
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/WARNING/URL	(#PCDATA)
	The URL for making another API request for the next batch of policy records.

Compliance Policy List: Glossary

XPath	element specifications / notes
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/GLOSSARY (ASSET_GROUP_LIST?, USER_LIST?)	
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/GLOSSARY/ASSET_GROUP_LIST (ASSET_GROUP+)	
	A list of asset groups assigned to policies in the policy list output.
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/GLOSSARY/ASSET_GROUP_LIST/ASSET_GROUP	
	(ID, TITLE, IP_SET?)
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/GLOSSARY/ASSET_GROUP_LIST /ASSET_GROUP/ID	
	(#PCDATA)
	An asset group ID for an asset group assigned to the policy.
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/GLOSSARY/ASSET_GROUP_LIST /ASSET_GROUP/TITLE	
	(#PCDATA)
	An asset group title for an asset group assigned to the policy.
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/GLOSSARY/ASSET_GROUP_LIST /ASSET_GROUP/IP_SET (IP|IP_RANGE)+	
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/GLOSSARY/ASSET_GROUP_LIST /ASSET_GROUP/IP_SET/IP (#PCDATA)	
	An IP address in an asset group that is assigned to the policy.
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/GLOSSARY/ASSET_GROUP_LIST /ASSET_GROUP/IP_SET/IP_RANGE (#PCDATA)	
	An IP address range in an asset group that is assigned to the policy.
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/GLOSSARY/USER_LIST (USER+)	
	A list of users who created or edited exceptions in compliance policies in the policy list output. For a policy that was edited, the user who most recently edited the exception is included in the output.
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/GLOSSARY/USER_LIST /USER	
	(USER_LOGIN, FIRST_NAME, LAST_NAME)
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/GLOSSARY/USER_LIST /USER (#PCDATA)	
	A user login ID.
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/GLOSSARY/USER_LIST /FIRST_NAME (#PCDATA)	
	The first name of the account user.
/COMPLIANCE_POLICY_LIST_OUTPUT/RESPONSE/GLOSSARY/USER_LIST /LAST_NAME (#PCDATA)	
	The last name of the account user.

Compliance Policy Export Output

The compliance policy export XML is returned from an API request for a compliance policy export. The DTD can be found at the following URL (where qualysapi.qualys.com is the API server URL where your account is located):

```
https://qualysapi.qualys.com/api/2.0/fo/compliance/policy/policy_export_output.dtd
```

DTD for Compliance Policy Export Output

A recent DTD is shown below.

```
<!-- QUALYS POLICY_EXPORT_OUTPUT DTD -->

<!ELEMENT POLICY_EXPORT_OUTPUT (REQUEST?, RESPONSE)>

<!ELEMENT REQUEST (DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?,
                   POST_DATA?)>
<!ELEMENT DATETIME (#PCDATA)>
<!ELEMENT USER_LOGIN (#PCDATA)>
<!ELEMENT RESOURCE (#PCDATA)>
<!ELEMENT PARAM_LIST (PARAM+)>
<!ELEMENT PARAM (KEY, VALUE)>
<!ELEMENT KEY (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>
<!-- if returned, POST_DATA will be urlencoded -->
<!ELEMENT POST_DATA (#PCDATA)>

<!ELEMENT RESPONSE (DATETIME, POLICY)>
<!ELEMENT POLICY (TITLE, DESCRIPTION?, LOCKED?, EXPORTED, COVER_PAGE?,
                  STATUS?, TECHNOLOGIES, SECTIONS)>
<!ELEMENT TITLE (#PCDATA)>
<!ELEMENT DESCRIPTION (#PCDATA)>
<!ELEMENT LOCKED (#PCDATA)>
<!ELEMENT EXPORTED (#PCDATA)>
<!ELEMENT COVER_PAGE (#PCDATA)>

<!ELEMENT SECTIONS (SECTION+)>
<!ATTLIST SECTIONS total CDATA #IMPLIED>
<!ELEMENT SECTION (NUMBER, HEADING, CONTROLS)>
<!ELEMENT NUMBER (#PCDATA)>
<!ELEMENT HEADING (#PCDATA)>

<!ELEMENT CONTROLS (CONTROL*)>
<!ATTLIST CONTROLS total CDATA #IMPLIED>
<!ELEMENT CONTROL (ID, CRITICALITY?, TECHNOLOGIES)>
<!ELEMENT ID (#PCDATA)>
```

```
<!ELEMENT STATUS (#PCDATA)>
<!ELEMENT CRITICALITY (LABEL, VALUE)>
<!ELEMENT LABEL (#PCDATA)>
<!ELEMENT TECHNOLOGIES (TECHNOLOGY*)>
<!ATTLIST TECHNOLOGIES total CDATA #IMPLIED>
<!ELEMENT TECHNOLOGY (ID, NAME?, EVALUATE?)>
<!ELEMENT NAME (#PCDATA)>
<!ELEMENT EVALUATE (CTRL*)>
<!ELEMENT CTRL (AND|OR|NOT|DP)+>
<!ELEMENT AND (AND|OR|NOT|DP)+>
<!ELEMENT OR (AND|OR|NOT|DP)+>
<!ELEMENT NOT (AND|OR|NOT|DP)+>
<!ELEMENT DP (K|OP|CD|L|V|FV)+>
<!ELEMENT K (#PCDATA)>
<!ELEMENT OP (#PCDATA)>
<!ELEMENT CD (#PCDATA)>
<!ELEMENT L (#PCDATA)>
<!ELEMENT V (#PCDATA)>
<!ELEMENT FV (#PCDATA)>
<!ATTLIST FV set CDATA #IMPLIED>
<!-- EOF -->
```

XPaths for Compliance Policy Export Output

This section describes the XPaths for the compliance policy export output.

Compliance Policy Export Output: Request

XPath	element specifications / notes
/POLICY_EXPORT_OUTPUT	(REQUEST?, RESPONSE)
/POLICY_EXPORT_OUTPUT/REQUEST	(DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?, POST_DATA?)
/OLICY_EXPORT_OUTPUT/REQUEST/DATETIME	(#PCDATA)
	The date and time of the request.
/POLICY_EXPORT_OUTPUT/REQUEST/USER_LOGIN	(#PCDATA)
	The user login ID of the user who made the request.
/POLICY_EXPORT_OUTPUT/REQUEST/RESOURCE	(#PCDATA)
	The resource specified for the request.
/POLICY_EXPORT_OUTPUT/REQUEST/PARAM_LIST	(PARAM+)
/POLICY_EXPORT_OUTPUT/REQUEST/PARAM_LIST/PARAM	(KEY, VALUE)
/POLICY_EXPORT_OUTPUT/REQUEST/PARAM_LIST/PARAM/KEY	(#PCDATA)
	An input parameter name.

XPath	element specifications / notes
/POLICY_EXPORT_OUTPUT/REQUEST/PARAM_LIST/PARAM/VALUE (#PCDATA)	
	An input parameter value.
/POLICY_EXPORT_OUTPUT/REQUEST/POST_DATA (#PCDATA)	
	The POST data, if any.

Compliance Policy Export Output: Response

XPath	element specifications / notes
/POLICY_EXPORT_OUTPUT/RESPONSE (REQUEST?, RESPONSE)	
/POLICY_EXPORT_OUTPUT/RESPONSE (DATETIME, POLICY)	
/POLICY_EXPORT_OUTPUT/RESPONSE/DATETIME (#PCDATA)	
	The date and time of the response.
/POLICY_EXPORT_OUTPUT/RESPONSE /POLICY	(TITLE, DESCRIPTION?, LOCKED? EXPORTED, COVER_PAGE?, STATUS?, TECHNOLOGIES, SECTIONS)
/POLICY_EXPORT_OUTPUT/RESPONSE/POLICY/TITLE (#PCDATA)	
	A compliance policy title.
/POLICY_EXPORT_OUTPUT/RESPONSE/POLICY/POLICY/DESCRIPTION (#PCDATA)	
	A compliance policy description.
/POLICY_EXPORT_OUTPUT/RESPONSE/POLICY/POLICY/LOCKED (#PCDATA)	
	A flag indicating that the policy is locked.
/POLICY_EXPORT_OUTPUT/RESPONSE/POLICY/EXPORTED (#PCDATA)	
	The date/time when the policy was exported.
/POLICY_EXPORT_OUTPUT/RESPONSE/POLICY/COVER_PAGE (#PCDATA)	
	Content for the cover page.
/POLICY_EXPORT_OUTPUT/RESPONSE/POLICY/STATUS (#PCDATA)	
	The current policy status: active or inactive.
/POLICY_EXPORT_OUTPUT/RESPONSE/POLICY/SECTIONS (SECTION+)	
	`total` is the total number of sections
/POLICY_EXPORT_OUTPUT/RESPONSE/POLICY/SECTIONS/SECTION (NUMBER, HEADING, CONTROLS)	
/POLICY_EXPORT_OUTPUT/RESPONSE/POLICY/SECTIONS/SECTION/NUMBER (#PCDATA)	
	A section number.
/POLICY_EXPORT_OUTPUT/RESPONSE/POLICY/SECTIONS/SECTION/HEADING (#PCDATA)	
	A section heading.
/POLICY_EXPORT_OUTPUT/RESPONSE/POLICY/SECTIONS/SECTION/CONTROLS (CONTROL*)	
	`total` is the total number of controls

XPath	element specifications / notes
/POLICY_EXPORT_OUTPUT/RESPONSE/POLICY/SECTIONS/SECTION/CONTROLS/CONTROL	(ID, CRITICALITY?, TECHNOLOGIES)
/POLICY_EXPORT_OUTPUT/RESPONSE/POLICY/SECTIONS/SECTION/CONTROLS/CONTROL/ID (#PCDATA)	A control ID.
/POLICY_EXPORT_OUTPUT/RESPONSE/POLICY/SECTIONS/SECTION/CONTROLS/CONTROL/CRITICALITY (LABEL, VALUE)	
/POLICY_EXPORT_OUTPUT/RESPONSE/POLICY/SECTIONS/SECTION/CONTROLS/CONTROL/CRITICALITY/LABEL (#PCDATA)	A criticality label (e.g. SERIOUS, CRITICAL, URGENT) assigned to the control.
/POLICY_EXPORT_OUTPUT/RESPONSE/POLICY/SECTIONS/SECTION/CONTROLS/CONTROL/CRITICALITY/VALUE (#PCDATA)	A criticality value (0-5) assigned to the control.
/POLICY_EXPORT_OUTPUT/RESPONSE/POLICY/SECTIONS/SECTION/CONTROLS/CONTROL/TECHNOLOGIES (TECHNOLOGY+)	**`total`** is the total number of technologies
/POLICY_EXPORT_OUTPUT/RESPONSE/POLICY/SECTIONS/SECTION/CONTROLS/CONTROL/TECHNOLOGIES /TECHNOLOGY	(ID, NAME?, EVALUATE?)
/POLICY_EXPORT_OUTPUT/RESPONSE/POLICY/SECTIONS/SECTION/CONTROLS/CONTROL/TECHNOLOGIES /TECHNOLOGY/ID (#PCDATA)	A technology ID.
/POLICY_EXPORT_OUTPUT/RESPONSE/POLICY/SECTIONS/SECTION/CONTROLS/CONTROL/TECHNOLOGIES /TECHNOLOGY/NAME (#PCDATA)	A technology name.
/POLICY_EXPORT_OUTPUT/RESPONSE/POLICY/SECTIONS/SECTION/CONTROLS/CONTROL/TECHNOLOGIES /TECHNOLOGY/EVALUATE (CTRL*)	The control evaluation logic.
attribute: checksum	This attribute is no longer returned in the XML output. However, you can still include it in policy export XML and import it into your account.
/POLICY_EXPORT_OUTPUT/RESPONSE/POLICY/SECTIONS/SECTION/CONTROLS/CONTROL/TECHNOLOGIES /TECHNOLOGY/EVALUATE/CTRL (AND|OR|NOT|DP)+	The root tag for control evaluation.
/POLICY_EXPORT_OUTPUT/RESPONSE/POLICY/SECTIONS/SECTION/CONTROLS/CONTROL/TECHNOLOGIES /TECHNOLOGY/EVALUATE/CTRL /AND (AND|OR|NOT|DP)+	Indicates a logical AND relationship between its children.
/POLICY_EXPORT_OUTPUT/RESPONSE/POLICY/SECTIONS/SECTION/CONTROLS/CONTROL/TECHNOLOGIES /TECHNOLOGY/EVALUATE/CTRL /OR (AND|OR|NOT|DP)+	Indicates a logical OR relationship between its children.
/POLICY_EXPORT_OUTPUT/RESPONSE/POLICY/SECTIONS/SECTION/CONTROLS/CONTROL/TECHNOLOGIES /TECHNOLOGY/EVALUATE/CTRL /NOT (AND|OR|NOT|DP)+	Indicates negation of evaluation logic respresented by its child tag.

XPath	element specifications / notes
/POLICY_EXPORT_OUTPUT/RESPONSE/POLICY/SECTIONS/SECTION/CONTROLS/CONTROL/TECHNOLOGIES /TECHNOLOGY/EVALUATE/CTRL /DP (K|OP|CD|L|V|FV)+	
	The evaluation logic for a data point in the compliance policy.
/POLICY_EXPORT_OUTPUT/RESPONSE/POLICY/SECTIONS/SECTION/CONTROLS/CONTROL/TECHNOLOGIES /TECHNOLOGY/EVALUATE/CTRL /DP/K (#PCDATA)	
	A service-defined, unique name for the data point.
/POLICY_EXPORT_OUTPUT/RESPONSE/POLICY/SECTIONS/SECTION/CONTROLS/CONTROL/TECHNOLOGIES /TECHNOLOGY/EVALUATE/CTRL /DP/OP (#PCDATA)	
	The operator option set in the compliance policy for the data point, if applicable. Possible values depending on the data type: ge | gt | le | lt | eq | ne | in | range | re | xre | xeq | no op. See "Operator Names" below.
/POLICY_EXPORT_OUTPUT/RESPONSE/POLICY/SECTIONS/SECTION/CONTROLS/CONTROL/TECHNOLOGIES /TECHNOLOGY/EVALUATE/CTRL /DP/CD (#PCDATA)	
	The cardinality option set in the compliance policy for the data point, if applicable. Possible values depending on the data type: contains | does not contain | matches | is contained in | intersect | match any | match all | match none | empty | not empty | no cd.
/POLICY_EXPORT_OUTPUT/RESPONSE/POLICY/SECTIONS/SECTION/CONTROLS/CONTROL/TECHNOLOGIES /TECHNOLOGY/EVALUATE/CTRL /DP/L (#PCDATA)	
	Identifies attributes of the data point that are locked and cannot be changed in the compliance policy. These data point attributes may be locked: OP (operator), CD (cardinality), V (expected value).
/POLICY_EXPORT_OUTPUT/RESPONSE/POLICY/SECTIONS/SECTION/CONTROLS/CONTROL/TECHNOLOGIES /TECHNOLOGY/EVALUATE/CTRL /DP/V (#PCDATA)	
	The user-provided "expected" value for the data point, as defined in the compliance policy.
/POLICY_EXPORT_OUTPUT/RESPONSE/POLICY/SECTIONS/SECTION/CONTROLS/CONTROL/TECHNOLOGIES /TECHNOLOGY/EVALUATE/CTRL /DP/FV (#PCDATA)	
	A fixed expected value for the data point in the compliance policy. A fixed value cannot be changed in the policy. It can only be selected/deselected.
attribute: set	**set** indicates whether the fixed value is selected in the compliance policy. When set=1 the fixed value is selected. When set=0 the fixed value is not selected.

Operator Names

See below for a description of operator names.

Operator	Description	Operator	Description
ge	greater than or equal to	in	in
gt	greater than	range	in range
le	less than or equal to	re	regular expression
lt	less than	xre	regular expression list

Operator	Description	Operator	Description
eq	equal to	xeq	string list
ne	not equal to	no op	no operator

Compliance Posture Information Output

The compliance posture information output XML is returned from an API request for compliance posture information. The DTD for the compliance posture info output XML is described below.

The DTD can be found at the following URL (where qualysapi.qualys.com is the API server URL where your account is located):

```
https://qualysapi.qualys.com/api/2.0/fo/compliance/posture/info/posture_info_list_output.dtd
```

DTD for Compliance Posture Information Output

A recent DTD for the compliance posture information output is shown below.

```
<!-- QUALYS POSTURE_INFO_LIST_OUTPUT DTD -->

<!ELEMENT POSTURE_INFO_LIST_OUTPUT (REQUEST?,RESPONSE)>

<!ELEMENT REQUEST (DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?,
                   POST_DATA?)>
<!ELEMENT DATETIME (#PCDATA)>
<!ELEMENT USER_LOGIN (#PCDATA)>
<!ELEMENT RESOURCE (#PCDATA)>
<!ELEMENT PARAM_LIST (PARAM+)>
<!ELEMENT PARAM (KEY, VALUE)>
<!ELEMENT KEY (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>
<!-- if returned, POST_DATA will be urlencoded -->
<!ELEMENT POST_DATA (#PCDATA)>

<!ELEMENT RESPONSE (DATETIME, ((INFO_LIST?, SUMMARY?, WARNING_LIST?,
                    GLOSSARY?) | POLICY+))>

<!ELEMENT POLICY (ID, DATETIME, INFO_LIST?, SUMMARY?, WARNING_LIST?,
                  GLOSSARY?)>

<!ELEMENT INFO_LIST (INFO+)>
<!ELEMENT INFO (ID, HOST_ID, CONTROL_ID, TECHNOLOGY_ID, INSTANCE?, STATUS,
                POSTURE_MODIFIED_DATE?, EXCEPTION?, EVIDENCE?)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT HOST_ID (#PCDATA)>
<!ELEMENT CONTROL_ID (#PCDATA)>
<!ELEMENT TECHNOLOGY_ID (#PCDATA)>
<!ELEMENT INSTANCE (#PCDATA)>
<!ELEMENT STATUS (#PCDATA)>
```

```
<!ELEMENT POSTURE_MODIFIED_DATE (#PCDATA)>
<!ELEMENT EXCEPTION (ASSIGNEE, STATUS, END_DATETIME?, CREATED?,
                     LAST_MODIFIED?, COMMENT_LIST?)>
<!ELEMENT ASSIGNEE (#PCDATA)>
<!ELEMENT END_DATETIME (#PCDATA)>
<!ELEMENT CREATED (BY, DATETIME)>
<!ELEMENT BY (#PCDATA)>
<!ELEMENT LAST_MODIFIED (BY, DATETIME)>
<!ELEMENT COMMENT_LIST (COMMENT+)>
<!ELEMENT COMMENT (DATETIME, BY, TEXT)>
<!ELEMENT TEXT (#PCDATA)>

<!ELEMENT EVIDENCE (BOOLEAN_EXPR, DPV_LIST?)>
<!ELEMENT BOOLEAN_EXPR (#PCDATA)>
<!ELEMENT DPV_LIST (DPV+)>
<!ELEMENT DPV (LABEL, (ERROR|V)+, TM_REF?)>
<!ATTLIST DPV lastUpdated CDATA #IMPLIED>

<!ELEMENT LABEL (#PCDATA)>
<!ELEMENT ERROR (#PCDATA)>
<!ELEMENT V (#PCDATA)>
<!ELEMENT TM_REF (#PCDATA)>

<!ELEMENT GLOSSARY (USER_LIST?, HOST_LIST, CONTROL_LIST?,
                    TECHNOLOGY_LIST?, DPD_LIST?, TP_LIST?, FV_LIST?,
                    TM_LIST?)>
<!ELEMENT USER_LIST (USER+)>
<!ELEMENT USER (USER_LOGIN, FIRST_NAME, LAST_NAME)>
<!ELEMENT FIRST_NAME (#PCDATA)>
<!ELEMENT LAST_NAME (#PCDATA)>

<!ELEMENT HOST_LIST (HOST+)>
<!ELEMENT HOST (ID, IP, TRACKING_METHOD, DNS?, NETBIOS?, OS?, OS_CPE?,
                LAST_VULN_SCAN_DATETIME?, LAST_COMPLIANCE_SCAN_DATETIME?,
                PERCENTAGE?)>
<!ELEMENT TRACKING_METHOD (#PCDATA)>
<!ELEMENT IP (#PCDATA)>
<!ATTLIST IP network_id  CDATA  #IMPLIED>
<!ELEMENT DNS (#PCDATA)>
<!ELEMENT NETBIOS (#PCDATA)>
<!ELEMENT OS (#PCDATA)>
<!ELEMENT OS_CPE (#PCDATA)>
<!ELEMENT LAST_VULN_SCAN_DATETIME (#PCDATA)>
<!ELEMENT LAST_COMPLIANCE_SCAN_DATETIME (#PCDATA)>
<!ELEMENT PERCENTAGE (#PCDATA)>

<!ELEMENT CONTROL_LIST (CONTROL+)>
<!ELEMENT CONTROL (ID, STATEMENT, CRITICALITY?, DEPRECATED?,
                   RATIONALE_LIST?)>
```

```
<!ELEMENT STATEMENT (#PCDATA)>
<!ELEMENT CRITICALITY (LABEL, VALUE)>
<!ELEMENT DEPRECATED (#PCDATA)>
<!ELEMENT RATIONALE_LIST (RATIONALE*)>
<!ELEMENT RATIONALE (TECHNOLOGY_ID, TEXT)>

<!ELEMENT TECHNOLOGY_LIST (TECHNOLOGY+)>
<!ELEMENT TECHNOLOGY (ID, NAME)>
<!ELEMENT NAME (#PCDATA)>

<!ELEMENT DPD_LIST (DPD+)>
<!ELEMENT DPD (LABEL, DESC)>
<!ELEMENT DESC (#PCDATA)>

<!ELEMENT TP_LIST (TP+)>
<!ELEMENT TP (LABEL, V+)>

<!ELEMENT FV_LIST (FV+)>
<!ELEMENT FV (LABEL, V*)>

<!ELEMENT TM_LIST (TM+)>
<!ELEMENT TM (LABEL, PAIR+)>
<!ELEMENT PAIR (K, V)>
<!ELEMENT K (#PCDATA)>

<!ELEMENT WARNING_LIST (WARNING+)>
<!ELEMENT WARNING (CODE?, TEXT, URL?)>
<!ELEMENT CODE (#PCDATA)>
<!ELEMENT URL (#PCDATA)>

<!ELEMENT SUMMARY (TOTAL_ASSETS, TOTAL_CONTROLS, CONTROL_INSTANCES)>
<!ELEMENT TOTAL_ASSETS (#PCDATA)>
<!ELEMENT TOTAL_CONTROLS (#PCDATA)>
<!ELEMENT CONTROL_INSTANCES (TOTAL, TOTAL_PASSED, TOTAL_FAILED,
                             TOTAL_ERROR, TOTAL_EXCEPTIONS)>
<!ELEMENT TOTAL (#PCDATA)>
<!ELEMENT TOTAL_PASSED (#PCDATA)>
<!ELEMENT TOTAL_FAILED (#PCDATA)>
<!ELEMENT TOTAL_ERROR (#PCDATA)>
<!ELEMENT TOTAL_EXCEPTIONS (#PCDATA)>

<!-- EOF -->
```

XPaths for Compliance Posture Information Output

This section describes the XPaths for the compliance posture information output.

Compliance Posture Information Output: Request

XPath	element specifications / notes
/POSTURE_INFO_LIST_OUTPUT	(REQUEST?, RESPONSE)
/POSTURE_INFO_LIST_OUTPUT/REQUEST	(DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?, POST_DATA?)
/POSTURE_INFO_LIST_OUTPUT/REQUEST/DATETIME	(#PCDATA) The date and time of the request.
/POSTURE_INFO_LIST_OUTPUT/REQUEST/USER_LOGIN	(#PCDATA) The user login ID of the user who made the request.
/POSTURE_INFO_LIST_OUTPUT/REQUEST/RESOURCE	(#PCDATA) The resource specified for the request.
/POSTURE_INFO_LIST_OUTPUT/REQUEST/PARAM_LIST	(PARAM+)
/POSTURE_INFO_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM	(KEY, VALUE)
/POSTURE_INFO_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM/KEY	(#PCDATA) An input parameter name.
/POSTURE_INFO_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM/VALUE	(#PCDATA) An input parameter value.
/POSTURE_INFO_LIST_OUTPUT/REQUEST/POST_DATA	(#PCDATA) The POST data, if any.

Compliance Posture Information Output: Response

XPath	element specifications / notes
/POSTURE_INFO_LIST_OUTPUT	(REQUEST?, RESPONSE)
/POSTURE_INFO_LIST_OUTPUT/RESPONSE	(DATETIME, ((INFO_LIST?, SUMMARY?, WARNING_LIST?, GLOSSARY?) | POLICY+))
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/DATETIME	(#PCDATA) The date and time of the response.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/POLICY	(ID, DATETIME, INFO_LIST?, SUMMARY?, WARNING_LIST?, GLOSSARY?)
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/POLICY/ID	(#PCDATA) The ID of a policy when “policy_ids” was specified.

XPath	element specifications / notes
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/POLICY/DATETIME (#PCDATA)	
	The date and time when the policy's posture info was collected from the API user's account.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/INFO_LIST (INFO+)	
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/INFO_LIST/INFO	
	(ID, HOST_ID, CONTROL_ID, TECHNOLOGY_ID, INSTANCE?, STATUS, EXCEPTION?, EVIDENCE?)
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/INFO_LIST/INFO/ID (#PCDATA)	
	A compliance posture info record ID.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/INFO_LIST/INFO/HOST_ID (#PCDATA)	
	A host ID for a compliance posture info record.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/INFO_LIST/INFO/CONTROL_ID (#PCDATA)	
	A control ID for a compliance posture info record.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/INFO_LIST/INFO/INSTANCE (#PCDATA)	
	An instance value for a compliance posture info record.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/INFO_LIST/INFO/STATUS (#PCDATA)	
	A compliance status for a compliance posture info record: Passed, Failed or Error. Error is returned only for a custom control in the case where an error occurred during control evaluation (and the ignore errors configuration option was not selected for the control).
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/INFO_LIST/INFO/EXCEPTION	
	(ASSIGNEE, STATUS, END_DATETIME?, CREATED?, LAST_MODIFIED?, COMMENT_LIST?)
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/INFO_LIST/INFO/EXCEPTION/ASSIGNEE (#PCDATA)	
	An assignee for an exception for a compliance posture info record.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/INFO_LIST/INFO/EXCEPTION/STATUS (#PCDATA)	
	The status of an exception for a compliance posture info record: Pending (approval), Accepted, Rejected or Expired.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/INFO_LIST/INFO/EXCEPTION/END_DATETIME (#PCDATA)	
	The date/time when an exception for a compliance posture info record expires (ends).
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/INFO_LIST/INFO/EXCEPTION/CREATED (BY, DATETIME)	
	The date/time when an exception for a compliance posture info record was created, and the user login ID of the user who created it.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/INFO_LIST/INFO/EXCEPTION/LAST_MODIFIED (BY, DATETIME)	
	The date/time when an exception for a compliance posture info record was last modified, and the user login ID of the user who modified it.

XPath	element specifications / notes
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/INFO_LIST/INFO/EXCEPTION/COMMENT_LIST (COMMENT+)	
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/INFO_LIST/INFO/EXCEPTION/COMMENT_LIST/COMMENT	(DATETIME, BY, TEXT)
	The date/time when comments were entered for an exception for a compliance posture info record, the user login ID of the user who entered these comments, and the text of the comments entered.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/INFO_LIST/INFO/EVIDENCE (BOOLEAN_EXPR, DPV_LIST?)	
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/INFO_LIST/INFO/EVIDENCE/BOOLEAN_EXPR (#PCDATA)	
	A Boolean expression string representing a data point rule for a control, which is used by the service to evaluate data point information gathered by the most recent compliance scan of the host. A data point rule is derived from a policy in the user's account. To understand why a posture info record has a Passed or Failed compliance status, take this boolean expression and plug in the data point "actual" values gathered from the most recent compliance scan in <DPV_LIST> and "expected" values as defined in the policy in <FV_LIST> or <TP_LIST>.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/INFO_LIST/INFO/EVIDENCE/DPV_LIST (DPV+)	
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/INFO_LIST/INFO/EVIDENCE/DPV_LIST/DPV	(LABEL, (ERROR|V)+, TM_REF?)
attribute: **lastUpdated**	**lastUpdated** is the most recent date/time the datapoint was scanned.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/INFO_LIST/INFO/EVIDENCE/DPV_LIST/DPV/LABEL	(#PCDATA)
	A label for a data point in the data point rule. This is a service-generated value in the format :dp_x such as :dp_1, :dp_2, :dp_3... These labels are not persistent and change each time an API call is made.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/INFO_LIST/INFO/EVIDENCE/DPV_LIST/DPV/ERROR	(#PCDATA)
	An error for a data point. The value NOT_FOUND is returned when a data point which is needed to evaluate a Boolean expression (in <BOOLEAN_EXPR>) was not detected on the host. When returned, no data point values are returned in <V> elements under <DPV_LIST>.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/INFO_LIST/INFO/EVIDENCE/DPV_LIST/DPV/V (#PCDATA)	
	A data point "actual" value, as returned from the most recent compliance scan.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/INFO_LIST/INFO/EVIDENCE/DPV_LIST/DPV/TM_REF	(#PCDATA)
	A translation context reference. This is a service-generated value in the format @tm_x such as @tm_1, @tm_2. @tm_3... These labels are not persistent and change each time an API call is made.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/INFO_LIST/SUMMARY	(TOTAL_ASSETS, TOTAL_CONTROLS, CONTROL_INSTANCES)
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/INFO_LIST/SUMMARY/TOTAL_ASSETS (#PCDATA)	
	Total number of hosts evaluated.

XPath	element specifications / notes
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/INFO_LIST/SUMMARY/TOTAL_CONTROLS (#PCDATA)	
	Total number of controls evaluated.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/INFO_LIST/SUMMARY/CONTROL_INSTANCES	
	(TOTAL, TOTAL_PASSED, TOTAL_FAILED, TOTAL_ERROR, TOTAL_EXCEPTIONS)
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/INFO_LIST/SUMMARY/CONTROL_INSTANCES/TOTAL (#PCDATA)	
	Total number of control instances evaluated.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/INFO_LIST/SUMMARY/CONTROL_INSTANCES/TOTAL_PASSED (#PCDATA)	
	Total number of control instances with passed status.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/INFO_LIST/SUMMARY/CONTROL_INSTANCES/TOTAL_FAILED (#PCDATA)	
	Total number of control instances with failed status
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/INFO_LIST/SUMMARY/CONTROL_INSTANCES/TOTAL_ERROR (#PCDATA)	
	Total number of control instances with error status.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/INFO_LIST/SUMMARY/CONTROL_INSTANCES/TOTAL_EXCEPTIONS (#PCDATA)	
	Total number of control instances with exceptions.

Compliance Posture Information Output: Glossary

XPath	element specifications / notes
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY	
	(USER_LIST?, HOST_LIST, CONTROL_LIST?, TECHNOLOGY_LIST?, DPD_LIST?, TP_LIST?, FV_LIST?, TM_LIST?)
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/USER_LIST (USER+)	
	A list of users who created, modified, or added comments to exceptions associated with compliance posture info records which are included in the posture information output.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/USER_LIST/USER	
	(USER_LOGIN, FIRST_NAME, LAST_NAME)
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/USER_LIST/USER (#PCDATA)	
	A user login ID associated with an exception in a posture info record.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/USER_LIST/FIRST_NAME (#PCDATA)	
	The first name of an account user associated with an exception in a posture info record.

XPath	element specifications / notes
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/USER_LIST/LAST_NAME (#PCDATA)	
	The last name of an account user associated with an exception in a posture info record.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/HOST_LIST (HOST+)	
	A list of hosts in compliance posture info records which are included in the posture list output.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/HOST_LIST/HOST	
	(ID, IP, TRACKING_METHOD, DNS?, NETBIOS?, OS?, OS_CPE?, LAST_VULN_SCAN_DATETIME?, LAST_COMPLIANCE_SCAN_DATETIME?, PERCENTAGE?)
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/HOST_LIST/HOST/ID (#PCDATA)	
	A host ID for a host in a posture info record.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/HOST_LIST/HOST/IP (#PCDATA)	
	An IP address for a host in a posture info record.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/HOST_LIST/HOST/TRACKING_METHOD (#PCDATA)	
	The tracking method for a host in a posture info record: IP, DNS or NETBIOS.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/HOST_LIST/HOST/DNS (#PCDATA)	
	The DNS user name for a host in a posture info record, when available.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/HOST_LIST/HOST/NETBIOS (#PCDATA)	
	The NetBIOS user name for a host in a posture info record, when available.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/HOST_LIST/HOST/OS (#PCDATA)	
	The operating system detected on a host in a posture info record, when available.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/HOST_LIST/HOST/OS_CPE (#PCDATA)	
	The OS CPE name assigned to the operating system detected on the host. (The OS CPE name appears only when the OS CPE feature is enabled for the subscription, and an authenticated scan was run on this host after enabling this feature.)
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/HOST_LIST/HOST/ LAST_VULN_SCAN_DATETIME (#PCDATA)	
	The date/time when a vulnerability scan was most recently launched on a host in a compliance posture info record.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/HOST_LIST/HOST/ LAST_COMPLIANCE_SCAN_DATETIME (#PCDATA)	
	The date/time when a compliance scan was most recently launched on a host in a compliance posture info record.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/HOST_LIST/HOST/PERCENTAGE (#PCDATA)	
	The percentage of controls that passed for the host. For example “85.71% (84 of 98)” mean 85.71% of the controls passed, 84 controls passed and 98 controls were evaluated).
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/CONTROL_LIST (CONTROL+)	
	A list of compliance controls in compliance posture info records which are included in the posture information output.

XPath	element specifications / notes
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/CONTROL_LIST/CONTROL	
	(ID, STATEMENT, CRITICALITY?, DEPRECATED?,RATIONALE_LIST?)
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/CONTROL_LIST/CONTROL/ID	(#PCDATA)
	A control ID.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/CONTROL_LIST/CONTROL/STATEMENT	
	(#PCDATA)
	A control statement.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/CONTROL_LIST/CONTROL/CRITICALITY	
	(LABEL, VALUE)
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/CONTROL_LIST/CONTROL/CRITICALITY/LABEL	
	(#PCDATA)
	A criticality label (e.g. SERIOUS, CRITICAL, URGENT) assigned to the control.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/CONTROL_LIST/CONTROL/CRITICALITY/VALUE	
	(#PCDATA)
	A criticality value (0-5) assigned to the control.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/CONTROL_LIST/CONTROL/DEPRECATED	
	(#PCDATA)
	The value 1 identifies a deprecated control. This element appears only for a deprecated control.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/CONTROL_LIST/CONTROL/RATIONALE _LIST	
	(RATIONALE*)
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/CONTROL_LIST/ CONTROL/RATIONALE_LIST/RATIONALE	
	(TECHNOLOGY_ID, TEXT)
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/CONTROL_LIST/ CONTROL/RATIONALE_LIST/RATIONALE/TECHNOLOGY_ID	(#PCDATA)
	An ID for a technology associated with a control's rationale..
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/CONTROL_LIST/ CONTROL/RATIONALE_LIST/RATIONALE/TEXT	(#PCDATA)
	A text description associated with a control's rationale.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/TECHNOLOGY_LIST	(TECHNOLOGY+)
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/TECHNOLOGY_LIST/TECHNOLOGY	(ID, NAME)
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/TECHNOLOGY_LIST/TECHNOLOGY/ID	
	(#PCDATA)
	An ID for a technology in a posture info record.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/TECHNOLOGY_LIST/TECHNOLOGY/NAME	
	(#PCDATA)
	A name for a technology in a compliance posture info record.

XPath	element specifications / notes
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/DPD_LIST (DPD+)	
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/DPD_LIST/DPD (LABEL, DESC)	
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/DPD_LIST/DPD/LABEL (#PCDATA)	
	A service-defined, internal label for a data point.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/DPD_LIST/DPD/DESC (#PCDATA)	
	A description for a data point, which corresponds to a data point label in a <LABEL> element.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/TP_LIST (TP+)	
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/TP_LIST/TP (LABEL, V+)	
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/TP_LIST/TP/LABEL (#PCDATA)	
	A label for a data point text pattern as defined in a policy. This is a service-generated value $tp_x such as $tp_1, $tp_2, $tp_3... The data point text pattern labels are not persistent and change each time an API call is made.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/TP_LIST/TP/V (#PCDATA)	
	A data point text pattern value in a policy.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/FV_LIST (FV+)	
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/FV_LIST/FV (LABEL, V*)	
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/FV_LIST/FV/LABEL (#PCDATA)	
	A label for a fixed value selection in a policy. This is a service-generated value #fv_x such as #fv_1, #fv_2, #fv_3... The data point fixed value labels are not persistent and change each time an API call is made.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/FV_LIST/FV/V (#PCDATA)	
	A data point fixed value selection in a policy.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/TM_LIST (TM+)	
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/TM_LIST/TM (LABEL, PAIR+)	
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/TM_LIST/TM/LABEL (#PCDATA)	
	A translation context reference. This is a service-generated value in the format @tm_x such as @tm_1, @tm_2. @tm_3... These labels are not persistent and change each time an API call is made.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/TM_LIST/TM/PAIR (K, V)	
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/TM_LIST/TM/PAIR/K (#PCDATA)	
	A translation context key in a mapping pair. This represents a raw, untranslated value returned by the scanning engine.
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/TM_LIST/PAIR/V (#PCDATA)	
	A translation context value in a mapping pair. This represents the meaning associated with the raw value in the mapping pair.

Compliance Posture Information Output: Warning

XPath	element specifications / notes
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/WARNING_LIST (WARNING+)	
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/WARNING (CODE?, TEXT, URL?)	
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/WARNING/CODE (#PCDATA)	
	A warning code. A warning code appears when the API request identifies more than 5,000 records (compliance posture info records).
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/WARNING/TEXT (#PCDATA)	
	A warning message. A warning message appears when the API request identifies more than 5,000 records (compliance posture info records).
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/WARNING/URL (#PCDATA)	
	A URL for making another API request for the next batch of records (compliance posture info records).

Compliance Evidence

This sections provides details about the compliance evidence information in the compliance posture information output (posture_info_output.dtd).

Boolean Expression

To understand why a control has a certain compliance status, take the boolean expression for a posture info record in this element:

```
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/INFO_LIST/INFO/EVIDENCE/BOOLEAN_EXPR
```

and plug in the data point "actual" values (such as :dp_1, :dp_2, :dp3, etc.) found in this element:

```
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/INFO_LIST/INFO/EVIDENCE/DPV_LIST
```

and text pattern "expected" values (such as $tp_1, $tp2, $tp3, etc.) found in this element:

```
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/TP_LIST
```

or fixed value selection "expected" values (such as #fv_1, #fv_2, #fv_3, etc.) found in this element:

```
/POSTURE_INFO_LIST_OUTPUT/RESPONSE/GLOSSARY/FV_LIST
```

Boolean Expression: Data Type Operators

The following operators may be used to construct a Boolean expression string. The operators are specific to the data type of the data point value.

For all operator descriptions: X is the "actual" data point value (in the most recent scan results) compared to Y which is the "expected" value (in a policy).

Operator	Description	Data Type	Example
>	X is greater than Y	Integer	:dp_1 > 3
<	X is less than Y	Integer	:dp_1 < 5
>=	X is greater than or equal to Y	Integer	:dp_2 >= 4
<=	X is less than or equal to Y	Integer	:dp_2 <= 2
==	X is equal to Y	Integer	:dp_1 == 2
!(X)	X not equal to Y	Integer	!(:dp_1 > 5)
matches	X matches Y	Regular Expression	:dp_4 matches $tp_1

Boolean Expression: Cardinality Operators

The following cardinality operators may be used to construct a Boolean expression string.

A cardinality operator is used to:

- Compare multiple "actual" values to a single "expected" value for a control
- Compare multiple "actual" values to multiple "expected" values for a control

For all cardinality operator descriptions: X is the "actual" data point value (in the most recent scan results) compared to Y which is the "expected" value (in a policy).

Cardinality Operator	Description	Data Type in List	Example
match_any	Match any X in Y	Integer Regular Expression	:dp_1 match_any $tp_5
match_all	Match all X in Y	Integer Regular Expression	:dp_1 match_all $tp_5
empty	X is empty	Integer Regular Expression	:dp_8 empty

Cardinality Operator	Description	Data Type in List	Example
not_empty	X is not empty	Integer Regular Expression	:dp_8 not_empty
contains	X contains all of Y	Integer Regular Expression	:dp_2 contains $tp_2
does_not_contain	X does not contain any of Y	Integer Regular Expression	:dp_2 does_not_contain $tp_1
intersect	Any value in X matches any value in Y	Integer Regular Expression	:dp_3 intersect $tp_5
matches	All values in X match all values in Y	Integer Regular Expression	:dp_3 matches $tp_2
is_contained_in	All values in X are contained in Y	Integer Regular Expression	:dp_9 is_contained_in $tp_3

Boolean Expression: Logical Grouping Operators

The following logical grouping operators may be used to construct a Boolean expression string.

For all operator descriptions: X is the "actual" data point value (in the most recent scan results) compared to Y which is the "expected" value (in a policy).

Operator	Description	Example
(X)	Evaluates subexpression X before evaluating anything outside of the parentheses	(:dp_1 > 5)
and	Combines two logical subexpressions (ANDed)	(:dp_1 < 4) and (:dp_1 > 8)
or	Combines two logical subexpressions (ORed)	(:dp_1 < 4) or (:dp_1 > 8)

Control Values

Certain values appear in data point control values, for example registry permissions and file/directory permissions. For information on control values, log into your Qualys account and search for "control values" in the help.

Compliance Policy Report

The policy report XML is returned when you download a saved policy report using the Report Share API. The DTD for the compliance policy report XML is described below and sample XML output for the three evaluation types is also provided.

The DTD can be found at the following URL (where qualysapi.qualys.com is the API server where your account is located):

```
https://qualysapi.qualys.com/compliance_policy_report.dtd
```

DTD for Compliance Policy Report

A recent DTD for the compliance policy report output (compliance_policy_report.dtd) is shown below.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- QUALYS COMPLIANCE POLICY REPORT DTD -->

<!ELEMENT COMPLIANCE_POLICY_REPORT (ERROR | (HEADER, (SUMMARY),
                                    (RESULTS)))>
<!ELEMENT ERROR (#PCDATA)>
<!ATTLIST ERROR number CDATA #IMPLIED>

<!ELEMENT HEADER (NAME, GENERATION_DATETIME, COMPANY_INFO, USER_INFO,
                  FILTERS)>
<!ELEMENT NAME (#PCDATA)>
<!ELEMENT GENERATION_DATETIME (#PCDATA)>

<!ELEMENT COMPANY_INFO (NAME, ADDRESS, CITY, STATE, COUNTRY, ZIP_CODE)>
<!ELEMENT ADDRESS (#PCDATA)>
<!ELEMENT CITY (#PCDATA)>
<!ELEMENT STATE (#PCDATA)>
<!ELEMENT COUNTRY (#PCDATA)>
<!ELEMENT ZIP_CODE (#PCDATA)>

<!ELEMENT USER_INFO (NAME, USERNAME, ROLE)>
<!ELEMENT USERNAME (#PCDATA)>
<!ELEMENT ROLE (#PCDATA)>

<!ELEMENT FILTERS (POLICY, POLICY_LOCKING?)>
<!ELEMENT POLICY (#PCDATA)>
<!ELEMENT POLICY_LOCKING (#PCDATA)>
<!ELEMENT SUMMARY (TOTAL_ASSETS, TOTAL_CONTROLS, CONTROL_INSTANCES,
                   CONTROLS_SUMMARY?, HOST_STATISTICS?)>
<!ELEMENT CONTROL_INSTANCES (TOTAL, TOTAL_PASSED, TOTAL_FAILED,
                             TOTAL_ERROR, TOTAL_EXCEPTIONS)>
```

```
<!ELEMENT TOTAL (#PCDATA)>
<!ELEMENT TOTAL_ASSETS (#PCDATA)>
<!ELEMENT TOTAL_CONTROLS (#PCDATA)>
<!ELEMENT TOTAL_PASSED (#PCDATA)>
<!ELEMENT TOTAL_FAILED (#PCDATA)>
<!ELEMENT TOTAL_ERROR (#PCDATA)>
<!ELEMENT TOTAL_EXCEPTIONS (#PCDATA)>

<!ELEMENT CONTROLS_SUMMARY (CONTROL_INFO*)>
<!ELEMENT CONTROL_INFO (ORDER, CONTROL_ID, STATEMENT, CRITICALITY?,
                        PERCENTAGE, DEPRECATED?)>
<!ELEMENT CONTROL_ID (#PCDATA)>
<!ELEMENT ORDER (#PCDATA)>
<!ELEMENT PERCENTAGE (#PCDATA)>
<!ELEMENT CRITICALITY (LABEL, VALUE)>
<!ELEMENT DEPRECATED (#PCDATA)>

<!ELEMENT RESULTS ( HOST_LIST, CHECKS?, DP_DESCRIPTIONS?) >
<!ELEMENT HOST_LIST (HOST*)>
<!ELEMENT HOST (TRACKING_METHOD, IP, DNS?, NETBIOS?, OPERATING_SYSTEM?,
               OS_CPE?, LAST_SCAN_DATE?,TOTAL_PASSED, TOTAL_FAILED,
               TOTAL_ERROR, TOTAL_EXCEPTIONS, ASSET_TAGS?, CONTROL_LIST,
               NETWORK?)>

<!ELEMENT CHECKS (CHECK*)>
<!ELEMENT CHECK (NAME, DP_NAME, EXPECTED, ACTUAL, PERMISSION_TRANSLATION?,
                 EXTENDED_EVIDENCE?, STATISTICS?)>
<!ELEMENT DP_NAME (#PCDATA)>
<!ELEMENT EXTENDED_EVIDENCE (#PCDATA)>
<!ELEMENT STATISTICS (SEARCH_DURATION, ERRORS?)>
<!ELEMENT EVALUATION (#PCDATA)>

<!ELEMENT EXPECTED (V*, CRITERIA?)>
<!ATTLIST EXPECTED logic CDATA #FIXED "OR">
<!ELEMENT CRITERIA (EVALUATION, V*)>
<!ELEMENT ACTUAL (V*)>
<!ELEMENT V (#PCDATA)>
<!ATTLIST ACTUAL lastUpdated CDATA #IMPLIED>


<!ELEMENT PERMISSION_TRANSLATION (PAIR+)>
<!ELEMENT PAIR (K, V)>
<!ELEMENT K (#PCDATA)>

<!ELEMENT DP_DESCRIPTIONS (DP*)>
<!ELEMENT DP (DP_NAME, DESCRIPTION, SCAN_PARAMETERS?)>
<!ELEMENT DESCRIPTION (#PCDATA) >

<!ELEMENT SCAN_PARAMETERS (PARAM*)>
```

```
<!ELEMENT PARAM (LABEL, VALUE)>
<!ELEMENT LABEL (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>

<!ELEMENT TRACKING_METHOD (#PCDATA)>
<!ELEMENT IP (#PCDATA)>
<!ELEMENT DNS (#PCDATA)>
<!ELEMENT NETBIOS (#PCDATA)>
<!ELEMENT OPERATING_SYSTEM (#PCDATA)>
<!ELEMENT OS_CPE (#PCDATA)>
<!ELEMENT LAST_SCAN_DATE (#PCDATA)>
<!ELEMENT ASSET_TAGS (ASSET_TAG*)>
<!ELEMENT ASSET_TAG (#PCDATA)>

<!ELEMENT CONTROL_LIST (CONTROL*)>
<!ELEMENT CONTROL (CID, STATEMENT, CRITICALITY?, CONTROL_REFERENCES?,
                   DEPRECATED?, RATIONALE?, INSTANCE?, STATUS,
                   EVALUATION_DATE?, EVIDENCE?, EXCEPTION?)>
<!ELEMENT CID (#PCDATA)>
<!ELEMENT STATEMENT (#PCDATA)>
<!ELEMENT CONTROL_REFERENCES (#PCDATA)>
<!ELEMENT RATIONALE (#PCDATA)>
<!ELEMENT STATUS (#PCDATA)>
<!ELEMENT EVALUATION_DATE (#PCDATA)>
<!ELEMENT INSTANCE (#PCDATA)>
<!ELEMENT EVIDENCE (#PCDATA)>
<!ELEMENT EXCEPTION (ASSIGNEE, STATUS, END_DATE, CREATED_BY, CREATED_DATE,
                     MODIFIED_BY, MODIFIED_DATE, COMMENT_LIST?)>
<!ELEMENT ASSIGNEE (#PCDATA)>
<!ELEMENT END_DATE (#PCDATA)>
<!ELEMENT CREATED_BY (#PCDATA)>
<!ELEMENT CREATED_DATE (#PCDATA)>
<!ELEMENT MODIFIED_BY (#PCDATA)>
<!ELEMENT MODIFIED_DATE (#PCDATA)>


<!ELEMENT NETWORK (#PCDATA)>
<!ELEMENT COMMENT_LIST (COMMENT+)>
<!ELEMENT COMMENT (DATETIME, BY, TEXT)>
<!ELEMENT TEXT (#PCDATA)>
<!ELEMENT DATETIME (#PCDATA)>
<!ELEMENT BY (#PCDATA)>

<!ELEMENT HOST_STATISTICS (HOST_INFO*)>
<!ELEMENT HOST_INFO (IP, DNS, NETBIOS, OPERATING_SYSTEM, LAST_SCAN_DATE,
                     PERCENTAGE, NETWORK?)>

<!ELEMENT SEARCH_DURATION (#PCDATA)>
<!ELEMENT ERRORS (#PCDATA)>
```

XPaths for Compliance Policy Report

The XPaths for the compliance policy report are described below.

XPath	element specifications / notes
/COMPLIANCE_POLICY_REPORT	(ERROR | (HEADER, (SUMMARY), (RESULTS)))
/COMPLIANCE_POLICY_REPORT/ERROR	(#PCDATA)
	An error message.
attribute: **number**	An error code, when available
/COMPLIANCE_POLICY_REPORT/HEADER	(NAME, GENERATION_DATETIME, COMPANY_INFO, USER_INFO, FILTERS)
/COMPLIANCE_POLICY_REPORT/HEADER/NAME	(#PCDATA)
	The report title as provided by the user at the time the report was generated. If a report title was not provided, then the report template title appears.
/COMPLIANCE_POLICY_REPORT/HEADER/GENERATION_DATETIME	(#PCDATA)
	The date and time when the report was generated.
/COMPLIANCE_POLICY_REPORT/HEADER/COMPANY_INFO	(NAME, ADDRESS, CITY, STATE, COUNTRY, ZIP_CODE)
	The user's company name and address, as defined in the user's account.
/COMPLIANCE_POLICY_REPORT/HEADER/USER_INFO	(NAME, USERNAME, ROLE)
/COMPLIANCE_POLICY_REPORT/HEADER/USER_INFO/NAME	(#PCDATA)
	The name of the user who generated the report.
/COMPLIANCE_POLICY_REPORT/HEADER/USER_INFO/USERNAME	(#PCDATA)
	The user login ID of the user who generated the report.
/COMPLIANCE_POLICY_REPORT/HEADER/USER_INFO/ROLE	(#PCDATA)
	The user role assigned to the user who generated the report: Manager, Unit Manager, Auditor, Scanner, or Reader.
/COMPLIANCE_POLICY_REPORT/HEADER/FILTERS	(POLICY, POLICY_LOCKING?)
/COMPLIANCE_POLICY_REPORT/HEADER/FILTERS/POLICY	(#PCDATA)
	The title of the policy included in the report.
/COMPLIANCE_POLICY_REPORT/HEADER/FILTERS/POLICY_LOCKING	(#PCDATA)
	The locking status for the policy included in the report: Locked or Unlocked.
/COMPLIANCE_POLICY_REPORT/SUMMARY	(TOTAL_ASSETS, TOTAL_CONTROLS, CONTROL_INSTANCES, CONTROLS_SUMMARY?, HOST_STATISTICS?)
/COMPLIANCE_POLICY_REPORT/SUMMARY/TOTAL_ASSETS	(#PCDATA)
	The number of hosts in the policy.
/COMPLIANCE_POLICY_REPORT/SUMMARY/TOTAL_CONTROLS	(#PCDATA)
	The number of controls in the policy.

XPath	element specifications / notes
/COMPLIANCE_POLICY_REPORT/SUMMARY/CONTROL_INSTANCES	(TOTAL, TOTAL_PASSED, TOTAL_FAILED, TOTAL_ERROR, TOTAL_EXCEPTIONS)
/COMPLIANCE_POLICY_REPORT/SUMMARY/CONTROL_INSTANCES/TOTAL (#PCDATA)	The number of control instances in the report (sum of passed and failed instances).
/COMPLIANCE_POLICY_REPORT/SUMMARY/CONTROL_INSTANCES/TOTAL_PASSED (#PCDATA)	The number of control instances with a Passed status in the report.
/COMPLIANCE_POLICY_REPORT/SUMMARY/CONTROL_INSTANCES/TOTAL_FAILED (#PCDATA)	The number of control instances with a Failed status in the report.
/COMPLIANCE_POLICY_REPORT/SUMMARY/CONTROL_INSTANCES/TOTAL_ERROR (#PCDATA)	The number of control instances with an Error status in the report. An error status is returned for a custom control only in the case where an error occurred during control evaluation (and the ignore errors configuration option was not selected for the control).
/COMPLIANCE_POLICY_REPORT/SUMMARY/CONTROL_INSTANCES/TOTAL_EXCEPTIONS (#PCDATA)	The number of approved and pending exceptions in the policy report.
/COMPLIANCE_POLICY_REPORT/SUMMARY/CONTROLS_SUMMARY (CONTROL_INFO*)	
/COMPLIANCE_POLICY_REPORT/SUMMARY/CONTROLS_SUMMARY/CONTROL_INFO	(ORDER, CONTROL_ID, STATEMENT, CRITICALITY?, PERCENTAGE, DEPRECATED?)
/COMPLIANCE_POLICY_REPORT/SUMMARY/CONTROLS_SUMMARY/CONTROL_INFO/ORDER	(#PCDATA)
	The order number of the control in the policy. Controls in section 1 are numbered 1.1, 1.2, 1.3, and so on. Controls in section 2 are numbered 2.1, 2.2, 2.3, and so on.
/COMPLIANCE_POLICY_REPORT/SUMMARY/CONTROLS_SUMMARY/CONTROL_INFO/CONTROL_ID	(#PCDATA)
	The control ID number assigned to the control.
/COMPLIANCE_POLICY_REPORT/SUMMARY/CONTROLS_SUMMARY/CONTROL_INFO/STATEMENT	(#PCDATA)
	The control statement that describes how a technology specific item should be implemented in the environment.
/COMPLIANCE_POLICY_REPORT/SUMMARY/CONTROLS_SUMMARY/CONTROL_INFO/CRITICALITY	(LABEL, VALUE)
/COMPLIANCE_POLICY_REPORT/SUMMARY/CONTROLS_SUMMARY/CONTROL_INFO/CRITICALITY/LABEL (#PCDATA)	A criticality label (e.g. SERIOUS, CRITICAL, URGENT) assigned to the control.
/COMPLIANCE_POLICY_REPORT/SUMMARY/CONTROLS_SUMMARY/CONTROL_INFO/CRITICALITY/VALUE (#PCDATA)	A criticality value (0-5) assigned to the control.

XPath	element specifications / notes
/COMPLIANCE_POLICY_REPORT/SUMMARY/CONTROLS_SUMMARY/CONTROL_INFO/PERCENTAGE	(#PCDATA)
	The percentage of hosts that passed for the control. For example, a value of "50% (3 of 6)" indicates that the control passed on 3 of the 6 hosts included in the report.
/COMPLIANCE_POLICY_REPORT/SUMMARY/CONTROLS_SUMMARY/CONTROL_INFO/DEPRECATED	(#PCDATA)
	The value 1 identifies a deprecated control. This element appears only for a deprecated control.
/COMPLIANCE_POLICY_REPORT/RESULTS	(HOST_LIST, CHECKS?, DP_DESCRIPTIONS?)
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST	(HOST*)
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST	(TRACKING_METHOD, IP, DNS?, NETBIOS?, OPERATING_SYSTEM?, OS_CPE?, LAST_SCAN_DATE?,TOTAL_PASSED, TOTAL_FAILED, TOTAL_ERROR, TOTAL_EXCEPTIONS, ASSET_TAGS?, CONTROL_LIST, NETWORK?)
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST/TRACKING_METHOD	(#PCDATA)
	The tracking method for the host: IP, DNS hostname or NetBIOS hostname.
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST/IP	(#PCDATA)
	The IP address for the host.
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST/DNS	(#PCDATA)
	The DNS hostname for the host, when available.
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST/NETBIOS	(#PCDATA)
	The NetBIOS hostname for the host, when available
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST/OPERATING_SYSTEM	(#PCDATA)
	The operating system detected on the host.
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST/OS_CPE	(#PCDATA)
	The OS CPE name assigned to the operating system detected on the host. (The OS CPE name appears only when the OS CPE feature is enabled for the subscription, and an authenticated scan was run on this host after enabling this feature.)
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST/LAST_SCAN_DATE	(#PCDATA)
	The date and time the host was last scanned for compliance.
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST/ASSET_TAGS	(ASSET_TAG*)
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST/ASSET_TAGS/ASSET_TAG	(#PCDATA)
	An asset tag assigned to the host when the Asset Tagging feature is enabled in the user's account.
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST/TOTAL_PASSED	(#PCDATA)
	The number of control in the policy that Passed on the host.
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST/TOTAL_FAILED	(#PCDATA)
	The number of controls in the policy that Failed on the host.

XPath	element specifications / notes
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST/TOTAL_ERROR (#PCDATA)	
	The number of custom controls in the policy that were assigned the Error status on the host, because an error during control evaluation.
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST/TOTAL_EXCEPTIONS (#PCDATA)	
	The number of approved and pending exceptions on the host. This includes control instances with the Failed and Error status.
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST/CONTROL_LIST (CONTROL*)	
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST/CONTROL_LIST/CONTROL	
	(CID, STATEMENT, CRITICALITY?, CONTROL_REFERENCES?, DEPRECATED?, RATIONALE?, INSTANCE?, STATUS, EVIDENCE?, EXCEPTION?)
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST/CONTROL_LIST/CONTROL/CID	
	(#PCDATA)
	The control ID number assigned to the control.
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST/CONTROL_LIST/CONTROL/STATEMENT (#PCDATA)	
	The control statement that describes how a technology specific item should be implemented in the environment.
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST/CONTROL_LIST/CONTROL/CRITICALITY	
	(LABEL, VALUE)
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST/CONTROL_LIST/CONTROL/CRITICALITY/LABEL (#PCDATA)	
	A criticality label (e.g. SERIOUS, CRITICAL, URGENT) assigned to the control.
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST/CONTROL_LIST/CONTROL/CRITICALITY/VALUE (#PCDATA)	
	A criticality value (0-5) assigned to the control.
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST/CONTROL_LIST/CONTROL/CONTROL_REFERENCES (#PCDATA)	
	User-defined references, added to the control using the Qualys user interface.
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST/CONTROL_LIST/CONTROL/DEPRECATED (#PCDATA)	
	The value 1 identifies a deprecated control. This element appears only for a deprecated control.
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST/CONTROL_LIST/CONTROL/RATIONALE	
	(#PCDATA)
	A rationale statement that describes how the control should be implemented for the technology.
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST/CONTROL_LIST/CONTROL/INSTANCE	
	(#PCDATA)
	Instance information for an Oracle host in this format: Oracle technology version:SID:port. For example: Oracle10:ora102030p:1521.

XPath	element specifications / notes
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST/CONTROL_LIST/CONTROL/STATUS	(#PCDATA)
	The status for the control on the host: Passed, Failed or Error.
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST/CONTROL_LIST/CONTROL/EVIDENCE	(#PCDATA)
	One or more data point checks that returned results for the control on the host during the scan. The data point checks appear as CHECK1, CHECK2, and so on, which correspond to the <NAME> element for each check.
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST/NETWORK (#PCDATA)	
	The network the host belongs to.
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST/CONTROL_LIST/CONTROL/EXCEPTION	(ASSIGNEE, STATUS, END_DATE, CREATED_BY, CREATED_DATE, MODIFIED_BY, MODIFIED_DATE, COMMENT_LIST?)
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST/CONTROL_LIST/CONTROL/EXCEPTION/ASSIGNEE (#PCDATA)	
	The name of the user who is assigned the exception.
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST/CONTROL_LIST/CONTROL/EXCEPTION/STATUS (#PCDATA)	
	The exception status: Pending, Accepted, Rejected and Expired.
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST/CONTROL_LIST/CONTROL/EXCEPTION/END_DATE (#PCDATA)	
	The date the exception is set to expire. Note that end dates are only relevant to Accepted exceptions.
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST/CONTROL_LIST/CONTROL/EXCEPTION/CREATED_BY (#PCDATA)	
	The user who requested the exception.
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST/CONTROL_LIST/CONTROL/EXCEPTION/CREATED_DATE (#PCDATA)	
	The date and time the exception was created.
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST/CONTROL_LIST/CONTROL/EXCEPTION/MODIFIED_BY (#PCDATA)	
	The user who last modified the exception.
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST/CONTROL_LIST/CONTROL/EXCEPTION/MODIFIED_DATE (#PCDATA)	
	The date and time the exception was modified.
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST/CONTROL_LIST/CONTROL/EXCEPTION/COMMENT_LIST (COMMENT+)	
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST/CONTROL_LIST/CONTROL/EXCEPTION/COMMENT_LIST/COMMENT (DATETIME, BY, TEXT)	

XPath	element specifications / notes
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST/CONTROL_LIST/CONTROL/EXCEPTION/COMMENT_LIST/COMMENT/DATETIME (#PCDATA)	
	The date and time when an action on the exception took place.
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST/CONTROL_LIST/CONTROL/EXCEPTION/COMMENT_LIST/COMMENT/BY (#PCDATA)	
	The user who performed the action on the exception.
/COMPLIANCE_POLICY_REPORT/RESULTS/HOST_LIST/HOST/CONTROL_LIST/CONTROL/EXCEPTION/COMMENT_LIST/COMMENT/TEXT (#PCDATA)	
	Comments entered by the user who performed the action on the exception.
/COMPLIANCE_POLICY_REPORT/RESULTS/CHECKS (CHECK*)	
/COMPLIANCE_POLICY_REPORT/RESULTS/CHECKS/CHECK	
	(NAME, DP_NAME, EXPECTED, ACTUAL, PERMISSION_TRANSLATION?, EXTENDED_EVIDENCE?, STATISTICS?)
/COMPLIANCE_POLICY_REPORT/RESULTS/CHECKS/CHECK/NAME (#PCDATA)	
	A service-defined tag assigned to each data point.
/COMPLIANCE_POLICY_REPORT/RESULTS/CHECKS/CHECK/DP_NAME (#PCDATA)	
	A service-defined, unique name for a data point. The data point name identifies whether the data point is custom, the type of check performed, and the data point ID number. For example: custom.reg_key_exist.1001660.
/COMPLIANCE_POLICY_REPORT/RESULTS/CHECKS/CHECK/EXPECTED (V*, CRITERIA?)	
	A data point “expected” value, as defined in the compliance policy. The “expected” value may include fixed value selections, user-customized evaluation criteria, or a combination of both.
attribute: **logic**	**logic** is a fixed value equal to “OR”. When present, the control will pass if the “actual” value matches any of the “expected” values defined for the data point in the policy. This includes fixed value selections and user-customized evaluation criteria.
/COMPLIANCE_POLICY_REPORT/RESULTS/CHECKS/CHECK/EXPECTED/V (#PCDATA)	
	A fixed value selected for the data point in the compliance policy.
/COMPLIANCE_POLICY_REPORT/RESULTS/CHECKS/CHECK/EXPECTED/CRITERIA (EVALUATION, V*)	
	User-customized evaluation criteria for the data point, as defined in the compliance policy.
/COMPLIANCE_POLICY_REPORT/RESULTS/CHECKS/CHECK/EXPECTED/CRITERIA/EVALUATION	
	(#PCDATA)
	A data point rule used by the service to evaluate data point information gathered by the most recent compliance scan of the host. The data point rule includes the operator and cardinality options set in the compliance policy for the data point, if applicable.
/COMPLIANCE_POLICY_REPORT/RESULTS/CHECKS/CHECK/EXPECTED/CRITERIA/V (#PCDATA)	
	The user-provided “expected” value for the data point, as defined in the compliance policy.

XPath	element specifications / notes
/COMPLIANCE_POLICY_REPORT/RESULTS/CHECKS/CHECK/ACTUAL (V*)	
	A data point "actual" value, as found by the service during the most recent scan.
attribute: **lastUpdated**	**lastUpdated** is the most recent date/time the datapoint was scanned.
/COMPLIANCE_POLICY_REPORT/RESULTS/CHECKS/CHECK/ACTUAL/V (#PCDATA)	
	A data point "actual" value, as returned from the most recent compliance scan on the host.
/COMPLIANCE_POLICY_REPORT/RESULTS/CHECKS/CHECK/PERMISSION_TRANSLATION (PAIR+)	
/COMPLIANCE_POLICY_REPORT/RESULTS/CHECKS/CHECK/PERMISSION_TRANSLATION/PAIR (K, V)	
/COMPLIANCE_POLICY_REPORT/RESULTS/CHECKS/CHECK/PERMISSION_TRANSLATION/PAIR/K (#PCDATA)	
	A translation context key in a mapping pair. This represents a raw, untranslated value returned by the scanning engine. Each key maps to a registry or file/directory permission returned in the "actual" value.
/COMPLIANCE_POLICY_REPORT/RESULTS/CHECKS/CHECK/PERMISSION_TRANSLATION/PAIR/V (#PCDATA)	
	A translation context value in a mapping pair. This represents the meaning associated with the raw value in the mapping pair.
/COMPLIANCE_POLICY_REPORT/RESULTS/CHECKS/CHECK/EXTENDED_EVIDENCE (#PCDATA)	
	Extended evidence includes additional findings/information collected during the evaluation of the control on the host. This may include results returned from queries made by the scanning engine when checking the control value.
/COMPLIANCE_POLICY_REPORT/RESULTS/CHECKS/CHECK/STATISTICS (SEARCH_DURATION, ERRORS?)	
/COMPLIANCE_POLICY_REPORT/RESULTS/CHECKS/CHECK/STATISTICS/SEARCH_DURATION (#PCDATA)	
	The duration of the directory search for this check.
/COMPLIANCE_POLICY_REPORT/RESULTS/CHECKS/CHECK/STATISTICS/ERRORS (#PCDATA)	
	Any errors reported by this directory search check.
/COMPLIANCE_POLICY_REPORT/RESULTS/DP_DESCRIPTIONS (DP*)	
/COMPLIANCE_POLICY_REPORT/RESULTS/DP_DESCRIPTIONS/DP (DP_NAME, DESCRIPTION, SCAN_PARAMETERS?)	
/COMPLIANCE_POLICY_REPORT/RESULTS/DP_DESCRIPTIONS/DP/DP_NAME (#PCDATA)	
	A service-defined, unique name for a data point. The data point name identifies whether the data point is custom, the type of check performed, and the data point ID number. For example: custom.reg_key_exist.1001660.
/COMPLIANCE_POLICY_REPORT/RESULTS/DP_DESCRIPTIONS/DP/DESCRIPTION (#PCDATA)	
	A user-provided description for the data point. (Applies to a custom control.)
/COMPLIANCE_POLICY_REPORT/RESULTS/DP_DESCRIPTIONS/DP/SCAN_PARAMETERS (PARAM*)	
/COMPLIANCE_POLICY_REPORT/RESULTS/DP_DESCRIPTIONS/DP/SCAN_PARAMETERS/PARAM (LABEL, VALUE)	
/COMPLIANCE_POLICY_REPORT/RESULTS/DP_DESCRIPTIONS/DP/SCAN_PARAMETERS/PARAM/LABEL	

XPath	element specifications / notes
	(#PCDATA)
	A service-defined label for a scan parameter: Registry Hive, Registry Key, NAME, File path, and Hash Type. (Only applies to a user-defined custom control.)
/COMPLIANCE_POLICY_REPORT/RESULTS/DP_DESCRIPTIONS/DP/SCAN_PARAMETERS/PARAM/VALUE	
	(#PCDATA)
	A value for a scan parameter, which corresponds to a scan parameter label in the <LABEL> element.
/COMPLIANCE_POLICY_REPORT/SUMMARY/HOST_STATISTICS	(HOST_INFO+)
/COMPLIANCE_POLICY_REPORT/SUMMARY/HOST_STATISTICS/HOST_INFO	(IP. DNS, NETBIOS, OPERATING_SYSTEM, LAST_SCAN_DATE, PERCENTAGE, NETWORK?)
/COMPLIANCE_POLICY_REPORT/SUMMARY/HOST_STATISTICS/HOST_INFO/IP	(#PCDATA)
	The host's IP address.
/COMPLIANCE_POLICY_REPORT/SUMMARY/HOST_STATISTICS/HOST_INFO/DNS	(#PCDATA)
	The host's DNS name.
/COMPLIANCE_POLICY_REPORT/SUMMARY/HOST_STATISTICS/HOST_INFO/NETBIOS	(#PCDATA)
	The host's NetBIOS hostname.
/COMPLIANCE_POLICY_REPORT/SUMMARY/HOST_STATISTICS/HOST_INFO/ OPERATING_SYSTEM (#PCDATA)	
	The host's NetBIOS hostname.
/COMPLIANCE_POLICY_REPORT/SUMMARY/HOST_STATISTICS/HOST_INFO/ LAST_SCAN_DATE (#PCDATA)	
	The most recent date the host was scanned.
/COMPLIANCE_POLICY_REPORT/SUMMARY/HOST_STATISTICS/HOST_INFO/PERCENTAGE	(#PCDATA)
	The percentage of controls that passed on the host.
/COMPLIANCE_POLICY_REPORT/SUMMARY/HOST_STATISTICS/HOST_INFO/NETWORK	(#PCDATA)
	The network the host belongs to.

Sample Compliance Policy Report XML Output

The compliance policy report XML includes three data point evaluation types: 1) user-customized evaluation criteria, 2) fixed value selection, and 3) a combination of user-customized evaluation criteria and fixed values. Sample XML output is provided below.

Sample 1: Only User-Customized Criteria (No Fixed Values)

A control that does not have any fixed values looks like this:

```
<CHECK>
    <NAME>CHECK14</NAME>
    <DP_NAME>auth.passwords.expirywarning</DP_NAME>
    <EXPECTED logic="OR">
        <CRITERIA>
            <EVALUATION><![CDATA[less than]]></EVALUATION>
            <V><![CDATA[ 14 ]]></V>
        </CRITERIA>
    </EXPECTED>
    <ACTUAL lastUpdated="2012-04-01T15:21:36Z">
        <V><![CDATA[14]]></V>
    </ACTUAL>
</CHECK>
```

Sample 2: Only Fixed Values (No User-Customized Criteria)

For controls that only allow fixed value selection (user must select/clear checkboxes in the policy editor), the evaluation looks like this:

```
<CHECK>
    <NAME>CHECK14</NAME>
    <DP_NAME>auth.passwords.expirywarning</DP_NAME>
    <EXPECTED logic="OR">
        <V><![CDATA[ Enabled]]></V>
        <V><![CDATA[ RegKey not found]]></V>
        <V><![CDATA[ RegSubKey not found]]></V>
    </EXPECTED>
    <ACTUAL lastUpdated="2012-04-01T15:21:36Z">
        <V><![CDATA[14]]></V>
    </ACTUAL>
</CHECK>
```

In this example, each fixed value checkbox selected in the policy is displayed in a separate <V> element under <EXPECTED>. Note that there is no <CRITERIA> element under <EXPECTED> because there is no user-customized evaluation criteria.

Sample 3: Fixed Values and User-Customized Criteria

For controls using the fixed values in addition to user-customized evaluation criteria, the evaluation looks like this:

```
<CHECK>
    <NAME>CHECK14</NAME>
    <DP_NAME>auth.passwords.expirywarning</DP_NAME>
    <EXPECTED logic="OR">
        <CRITERIA>
            <EVALUATION><![CDATA[less than]]></EVALUATION>
            <V><![CDATA[14]]></V>
        </CRITERIA>
        <V><![CDATA[ RegSubKey not found]]></V>
    </EXPECTED>
    <ACTUAL lastUpdated="2012-04-01T15:21:36Z">
        <V><![CDATA[14]]></V>
    </ACTUAL>
</CHECK>
```

In this example, the <EXPECTED> element is used to display both the fixed value checkbox selections and the user-provided evaluation criteria (less than operator + value 14).

Compliance Authentication Report

The authentication report XML is returned when you download a saved authentication report using the Qualys user interface.

The DTD can be found at the following URL (where qualysapi.qualys.com is the API server URL where your account is located):

`https://qualysapi.qualys.com/compliance_authentication_report.dtd`

DTD for Compliance Authentication Report

A recent DTD for the compliance authentication report output (compliance_authentication_report.dtd) is shown below.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- QUALYS COMPLIANCE AUTHENTICATION REPORT DTD -->

<!ELEMENT COMPLIANCE_AUTHENTICATION_REPORT (ERROR | (HEADER,
                              (BUSINESS_UNIT_LIST | ASSET_GROUP_LIST |
                               ASSET_TAG_LIST)))>
<!ELEMENT ERROR (#PCDATA)>
<!ATTLIST ERROR number CDATA #IMPLIED>

<!ELEMENT HEADER (NAME, GENERATION_DATETIME, COMPANY_INFO, USER_INFO,
                  FILTERS)>
<!ELEMENT NAME (#PCDATA)>
<!ELEMENT GENERATION_DATETIME (#PCDATA)>

<!ELEMENT COMPANY_INFO (NAME, ADDRESS, CITY, STATE, COUNTRY, ZIP_CODE)>
<!ELEMENT ADDRESS (#PCDATA)>
<!ELEMENT CITY (#PCDATA)>
<!ELEMENT STATE (#PCDATA)>
<!ELEMENT COUNTRY (#PCDATA)>
<!ELEMENT ZIP_CODE (#PCDATA)>

<!ELEMENT USER_INFO (NAME, USERNAME, ROLE)>
<!ELEMENT USERNAME (#PCDATA)>
<!ELEMENT ROLE (#PCDATA)>
<!ELEMENT FILTERS (BUSINESS_UNIT_LIST | ASSET_GROUP_LIST)>
                   ASSET_TAG_LIST)>

<!ELEMENT BUSINESS_UNIT_LIST (BUSINESS_UNIT*)>
<!ELEMENT BUSINESS_UNIT (NAME | (NAME, AUTH_PASSED, AUTH_INSUFFICIENT,
                         AUTH_TOTAL, PASSED_PERCENTAGE,
                         TECHNOLOGY_LIST?))>
<!ELEMENT AUTH_PASSED (#PCDATA)>
```

```
<!ELEMENT AUTH_INSUFFICIENT (#PCDATA)>
<!ELEMENT AUTH_TOTAL (#PCDATA)>
<!ELEMENT PASSED_PERCENTAGE (#PCDATA)>

<!ELEMENT ASSET_TAG_LIST ((INCLUDED_TAGS, EXCLUDED_TAGS?) | ASSET_TAG)>
<!ELEMENT ASSET_TAG ((INCLUDED_TAGS, EXCLUDED_TAGS?) | (INCLUDED_TAGS,
                     EXCLUDED_TAGS?, AUTH_PASSED, AUTH_INSUFFICIENT,
                     AUTH_TOTAL, PASSED_PERCENTAGE, TECHNOLOGY_LIST?))>
<!ELEMENT INCLUDED_TAGS (TAG_ITEM+)>
<!ATTLIST INCLUDED_TAGS scope (any|all) #REQUIRED>
<!ELEMENT EXCLUDED_TAGS (TAG_ITEM+)>
<!ATTLIST EXCLUDED_TAGS scope (any|all) #REQUIRED>
<!ELEMENT TAG_ITEM (#PCDATA)>

<!ELEMENT ASSET_GROUP_LIST (ASSET_GROUP*)>
<!ELEMENT ASSET_GROUP (NAME | (NAME, AUTH_PASSED, AUTH_INSUFFICIENT,
                      AUTH_TOTAL, PASSED_PERCENTAGE, TECHNOLOGY_LIST?))>

<!ELEMENT TECHNOLOGY_LIST (TECHNOLOGY*)>
<!ELEMENT TECHNOLOGY (NAME, HOST_LIST)>
<!ELEMENT HOST_LIST (HOST*)>
<!ELEMENT HOST (TRACKING_METHOD, IP, DNS?, NETBIOS?, HOST_TECHNOLOGY?,
               INSTANCE?, STATUS, CAUSE?)>
<!ELEMENT TRACKING_METHOD (#PCDATA)>
<!ELEMENT IP (#PCDATA)>
<!ELEMENT DNS (#PCDATA)>
<!ELEMENT HOST_TECHNOLOGY (#PCDATA)>
<!ELEMENT NETBIOS (#PCDATA)>
<!ELEMENT INSTANCE (#PCDATA)>
<!ELEMENT STATUS (#PCDATA)>
<!ELEMENT CAUSE (#PCDATA)>
```

XPaths for Compliance Authentication Report

The XPaths for the compliance authentication report are described below.

XPath	element specifications / notes
/COMPLIANCE_AUTHENTICATION_REPORT	(ERROR | (HEADER, (BUSINESS_UNIT_LIST | ASSET_TAG_LIST ASSET_GROUP_LIST)))
/COMPLIANCE_AUTHENTICATION_REPORT/ERROR (#PCDATA)	An error message.
attribute: **number**	An error code, when available
/COMPLIANCE_AUTHENTICATION_REPORT/HEADER	(NAME, GENERATION_DATETIME, COMPANY_INFO, USER_INFO, FILTERS)

XPath	element specifications / notes
/COMPLIANCE_AUTHENTICATION_REPORT/HEADER/NAME (#PCDATA)	
	The report title as provided by the user at the time the report was generated. If a report title was not provided, then "Authentication Report" appears.
/COMPLIANCE_AUTHENTICATION_REPORT/HEADER/GENERATION_DATETIME (#PCDATA)	
	The date and time when the report was generated.
/COMPLIANCE_AUTHENTICATION_REPORT/HEADER/COMPANY_INFO	
	(NAME, ADDRESS, CITY, STATE, COUNTRY, ZIP_CODE)
	The user's company name and address, as defined in the user's account.
/COMPLIANCE_AUTHENTICATION_REPORT/HEADER/USER_INFO (NAME, USERNAME, ROLE)	
/COMPLIANCE_AUTHENTICATION_REPORT/HEADER/USER_INFO/NAME (#PCDATA)	
	The name of the user who generated the report.
/COMPLIANCE_AUTHENTICATION_REPORT/HEADER/USER_INFO/USERNAME (#PCDATA)	
	The user login ID of the user who generated the report.
/COMPLIANCE_AUTHENTICATION_REPORT/HEADER/USER_INFO/ROLE (#PCDATA)	
	The user role assigned to the user who generated the report: Manager, Unit Manager, Auditor, Scanner, or Reader.
/COMPLIANCE_AUTHENTICATION_REPORT/HEADER/FILTERS	
	(BUSINESS_UNIT_LIST | ASSET_TAG_LIST | ASSET_GROUP_LIST)
/COMPLIANCE_AUTHENTICATION_REPORT/HEADER/FILTERS/BUSINESS_UNIT_LIST (BUSINESS_UNIT*)	
	The business units included in the report source. (The report source either included a list of business units or a list of asset groups, not both.)
/COMPLIANCE_AUTHENTICATION_REPORT/HEADER/FILTERS/BUSINESS_UNIT_LIST/BUSINESS_UNIT	
	(NAME | (NAME, AUTH_PASSED, AUTH_INSUFFICIENT, AUTH_TOTAL, PASSED_PERCENTAGE, TECHNOLOGY_LIST?)))
	Host information for a business unit.
/COMPLIANCE_AUTHENTICATION_REPORT/HEADER/FILTERS/ASSET_TAG_LIST (ASSET_TAGS)	
	(INCLUDED_TAGS, EXCLUDED_TAGS)
/COMPLIANCE_AUTHENTICATION_REPORT/HEADER/FILTERS/ASSET_TAG_LIST/INCLUDED_TAGS/	
	ASSET_TAG (#PCDATA)
	The list of asset tags included in the report source. The scope "all" means hosts matching all tags; scope "any" means hosts matching at least one of the tags.
/COMPLIANCE_AUTHENTICATION_REPORT/HEADER/FILTERS/ASSET_TAG_LIST/EXCLUDED_TAGS/	
	ASSET_TAG (#PCDATA)
	The list of asset tags excluded from the report source. The scope "all" means hosts matching all tags; scope "any" means hosts matching at least one of the tags.
/COMPLIANCE_AUTHENTICATION_REPORT/HEADER/FILTERS/ASSET_GROUP_LIST (ASSET_GROUP*)	
	The asset groups included in the report source. (The report source either included a list of business units or a list of asset groups, not both.)

XPath	element specifications / notes
/COMPLIANCE_AUTHENTICATION_REPORT/HEADER/FILTERS/ASSET_GROUP_LIST /ASSET_GROUP	
	(NAME | (NAME, AUTH_PASSED, AUTH_INSUFFICIENT, AUTH_TOTAL, PASSED_PERCENTAGE, TECHNOLOGY_LIST?)))
	Host information for an asset group.
/COMPLIANCE_AUTHENTICATION_REPORT/HEADER/FILTERS/{type_list}/{type}/NAME	
	(#PCDATA)
	The name of the asset group or business unit.
/COMPLIANCE_AUTHENTICATION_REPORT/HEADER/FILTERS/{type_list}/{type}/AUTH_PASSED	
	(#PCDATA)
	The number of scanned hosts that passed authentication for the asset group or business unit, meaning that the scanning engine was able to successfully authenticate to the hosts.
/COMPLIANCE_AUTHENTICATION_REPORT/HEADER/FILTERS/{type_list}/{type}/AUTH_INSUFFICIENT	
	(#PCDATA)
	The number of scanned hosts that passed with insufficient privileges for the asset group or business unit, meaning that the scanning engine was able to authenticate to the hosts but there were insufficient privileges to perform posture evaluation.
/COMPLIANCE_AUTHENTICATION_REPORT/HEADER/FILTERS/{type_list}/{type}/AUTH_TOTAL	
	(#PCDATA)
	The total number of scanned hosts for the asset group or business unit.
/COMPLIANCE_AUTHENTICATION_REPORT/HEADER/FILTERS/{type_list}/{type}/PASSED_PERCENTAGE	
	(#PCDATA)
	The percentage of scanned hosts that passed for the asset group or business unit. For example, if 6 hosts were scanned and the scanning engine was able to authenticate to 3 of the scanned hosts, then 50 appears for 50% passed.
/COMPLIANCE_AUTHENTICATION_REPORT/HEADER/FILTERS/{type_list}/{type}/TECHNOLOGY_LIST	
	(TECHNOLOGY*)
/COMPLIANCE_AUTHENTICATION_REPORT/HEADER/FILTERS/{type_list}/{type}/TECHNOLOGY_LIST/ TECHNOLOGY (NAME, HOST_LIST)	
/COMPLIANCE_AUTHENTICATION_REPORT/HEADER/FILTERS/{type_list}/{type}/TECHNOLOGY_LIST/ TECHNOLOGY/NAME (#PCDATA)	
	The authentication type: Windows, SSH, Oracle, Oracle Listener, SNMP, or MS SQL.
/COMPLIANCE_AUTHENTICATION_REPORT/HEADER/FILTERS/{type_list}/{type}/TECHNOLOGY_LIST/ TECHNOLOGY/HOST_LIST (HOST*)	
/COMPLIANCE_AUTHENTICATION_REPORT/HEADER/FILTERS/{type_list}/{type}/TECHNOLOGY_LIST/ TECHNOLOGY/HOST_LIST/HOST	
	(TRACKING_METHOD, IP, DNS?, NETBIOS?, HOST_TECHNOLOGY?, INSTANCE?, STATUS, CAUSE?)

XPath	element specifications / notes
/COMPLIANCE_AUTHENTICATION_REPORT/HEADER/FILTERS/{type_list}/{type}/TECHNOLOGY_LIST/TECHNOLOGY/HOST_LIST/HOST/TRACKING_METHOD (#PCDATA)	
	The tracking method assigned to the host: IP, DNS, or NETBIOS.
/COMPLIANCE_AUTHENTICATION_REPORT/HEADER/FILTERS/{type_list}/{type}/TECHNOLOGY_LIST/TECHNOLOGY/HOST_LIST/HOST/IP (#PCDATA)	
	The IP address for the host.
/COMPLIANCE_AUTHENTICATION_REPORT/HEADER/FILTERS/{type_list}/{type}/TECHNOLOGY_LIST/TECHNOLOGY/HOST_LIST/HOST/DNS (#PCDATA)	
	The DNS hostname for the host, when available.
/COMPLIANCE_AUTHENTICATION_REPORT/HEADER/FILTERS/{type_list}/{type}/TECHNOLOGY_LIST/TECHNOLOGY/HOST_LIST/HOST/NETBIOS (#PCDATA)	
	The NetBIOS hostname for the host, when available.
/COMPLIANCE_AUTHENTICATION_REPORT/HEADER/FILTERS/{type_list}/{type}/TECHNOLOGY_LIST/TECHNOLOGY/HOST_LIST/HOST/HOST_TECHNOLOGY (#PCDATA)	
	The compliance technology the host's operating system is matched to.
/COMPLIANCE_AUTHENTICATION_REPORT/HEADER/FILTERS/{type_list}/{type}/TECHNOLOGY_LIST/TECHNOLOGY/HOST_LIST/HOST/INSTANCE (#PCDATA)	
	If the compliance information applies to a technology version on the host, like an Oracle version, instance information appears in this format: Port <number>, SID <value>. For example: Port 1521, SID ora010203p.
/COMPLIANCE_AUTHENTICATION_REPORT/HEADER/FILTERS/{type_list}/{type}/TECHNOLOGY_LIST/TECHNOLOGY/HOST_LIST/HOST/STATUS (#PCDATA)	
	The host's authentication status: Passed, Failed, or Passed*. Passed* indicates that authentication to the host was successful but the login account had insufficient privileges.
/COMPLIANCE_AUTHENTICATION_REPORT/HEADER/FILTERS/{type_list}/{type}/TECHNOLOGY_LIST/TECHNOLOGY/HOST_LIST/HOST/CAUSE (#PCDATA)	
	Additional information for a host with a Failed or Passed* status. This may include the login ID used during the authentication attempt.

Compliance Scorecard Report

The compliance scorecard report XML is returned when you download a saved report using the Qualys user interface.

The DTD can be found at the following URL (where qualysapi.qualys.com is the API server URL where your account is located):

```
https://qualysapi.qualys.com/compliance_scorecard_report.dtd
```

DTD for Compliance Scorecard Report

A recent DTD for the compliance scorecard report output is shown below.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- QUALYS COMPLIANCE SCORECARD REPORT DTD -->

<!ELEMENT COMPLIANCE_SCORECARD_REPORT (ERROR | (HEADER, (SUMMARY),
                                      (DETAILS)))>
<!ELEMENT ERROR (#PCDATA|COUNT|PERCENT)*>
<!ATTLIST ERROR number CDATA #IMPLIED>

<!ELEMENT HEADER (REPORT_TYPE, GENERATION_DATETIME)>
<!ELEMENT SUMMARY (ABOUT_REPORT, REPORT_SETTINGS, REPORT_DISCOVERIES)>
<!ELEMENT ABOUT_REPORT (REPORT_TYPE, CREATED, USER_NAME, LOGIN_NAME,
                        USER_ROLE, COMPANY_INFO)>
<!ELEMENT COMPANY_INFO (NAME, ADDRESS, CITY, STATE, COUNTRY, ZIP_CODE)>
<!ELEMENT REPORT_SETTINGS (TEMPLATE, NUMBER_OF_POLICIES,
                           REPORT_TIMEFRAME, ASSET_GROUPS*, ASSET_TAGS*,
                           CRITICALITY*)>
<!ELEMENT REPORT_DISCOVERIES (OVERALL_COMPLIANCE, BY_CONTROL, BY_HOSTS,
                              BY_TECHNOLOGY, BY_CRITICALITY*)>
<!ELEMENT ASSET_GROUPS (ASSET_GROUP_NAME)+>
<!ELEMENT ASSET_TAGS ((INCLUDED_TAGS, EXCLUDED_TAGS?) | ASSET_TAG?)>
<!ELEMENT OVERALL_COMPLIANCE (OVERALL_COMPLIANCE_PERCENT, UNIQUE_POLICES,
                              PASSED, FAILED, ERROR)>
<!ELEMENT BY_CONTROL (TOTAL_CONTROL_DETECTED, CHANGED_CONTROL, PASSED,
                      FAILED, ERROR)>
<!ELEMENT PASSED (COUNT, PERCENT)>
<!ELEMENT FAILED (COUNT, PERCENT)>
<!ELEMENT BY_HOSTS (TOTAL_HOSTS_IN_POLICIES, SCANNED_HOSTS, CHANGED)>
<!ELEMENT BY_TECHNOLOGY ((TOTAL_TECHNOLOGY, CHANGED_TECHNOLOGY,
                          TECHNOLOGY*)|(TECHNOLOGY+))>
<!ELEMENT TECHNOLOGY
(#PCDATA|NAME|CONTROL_INSTANCES|COUNT|PERCENT|PASSED_TOTAL|PASSED_CHANGED
|FAILED_TOTAL|FAILED_CHANGED|ERROR_TOTAL|ERROR_CHANGED|COMPLIANCE)*>
<!ELEMENT DETAILS (COMPLIANCE_BY_POLICY*, COMPLIANCE_BY_ASSET_GROUP*,
```

```
                    COMPLIANCE_BY_ASSET_TAG*, COMPLIANCE_BY_TECHNOLOGY*,
                    COMPLIANCE_BY_CRITICALITY*, TOP_HOST_WITH_CHANGES*,
                    TOP_CONTROLS_WITH_CHANGES*,
                    FAILED_CONTROLS_BY_CRITICALITY*)>
<!ELEMENT COMPLIANCE_BY_POLICY (DETAIL_DATE, BY_POLICY*,
                                BY_POLICY_ASSET_GROUP*,
                                BY_POLICY_ASSET_TAG*,
                                BY_POLICY_TECHNOLOGY*)>
<!ELEMENT COMPLIANCE_BY_ASSET_GROUP (DETAIL_DATE, BY_ASSET_GROUP*,
                                     BY_ASSET_GROUP_POLICY*,
                                     BY_ASSET_GROUP_TECHNOLOGY*)>
<!ELEMENT COMPLIANCE_BY_ASSET_TAG (DETAIL_DATE, BY_ASSET_TAG*,
                                   BY_ASSET_TAG_POLICY*,
                                   BY_ASSET_TAG_TECHNOLOGY*)>
<!ELEMENT COMPLIANCE_BY_TECHNOLOGY (DETAIL_DATE, BY_TECHNOLOGY)>
<!ELEMENT COMPLIANCE_BY_CRITICALITY (DETAIL_DATE, BY_CRITICALITY*,
                                     BY_CRITICALITY_POLICY*,
                                     BY_CRITICALITY_ASSET_GROUP*,
                                     BY_CRITICALITY_ASSET_TAG*,
                                     BY_CRITICALITY_TECHNOLOGY*)>
<!ELEMENT TOP_HOST_WITH_CHANGES (TOP, CHANGED_TO_PASS, CHANGED_TO_FAIL,
                                 CHANGED_TO_ERROR)>
<!ELEMENT TOP_CONTROLS_WITH_CHANGES (TOP, CHANGED_TO_PASS,
                                     CHANGED_TO_FAIL, CHANGED_TO_ERROR)>
<!ELEMENT FAILED_CONTROLS_BY_CRITICALITY (FAILED_CONTROLS*)>

<!ELEMENT BY_POLICY (POLICY+)>
<!ELEMENT BY_POLICY_ASSET_GROUP (POLICY+)>
<!ELEMENT BY_POLICY_ASSET_TAG (POLICY+)>
<!ELEMENT BY_POLICY_TECHNOLOGY (POLICY+)>
<!ELEMENT BY_ASSET_GROUP (ASSET_GROUP+)>
<!ELEMENT BY_ASSET_TAG (ASSET_TAG+)>
<!ELEMENT BY_ASSET_GROUP_POLICY (ASSET_GROUP+)>
<!ELEMENT BY_ASSET_TAG_POLICY (ASSET_TAG+)>
<!ELEMENT BY_ASSET_GROUP_TECHNOLOGY (ASSET_GROUP+)>
<!ELEMENT BY_ASSET_TAG_TECHNOLOGY (ASSET_TAG+)>
<!ELEMENT BY_CRITICALITY (TOTAL_FAILED_CONTROLS*,
                          TOTAL_FAILED_CONTROLS_CHANGED*, CRITICALITY*)>
<!ELEMENT BY_CRITICALITY_POLICY (CRITICALITY*)>
<!ELEMENT BY_CRITICALITY_ASSET_GROUP (CRITICALITY*)>
<!ELEMENT BY_CRITICALITY_ASSET_TAG (CRITICALITY*)>
<!ELEMENT BY_CRITICALITY_TECHNOLOGY (CRITICALITY*)>
<!ELEMENT FAILED_CONTROLS (CRITICALITY*)>

<!ELEMENT POLICY (POLICY_TITLE, ASSET_GROUP?, ASSET_TAG?, TECHNOLOGY?,
                  CONTROL_INSTANCES, HOSTS_TOTAL, HOSTS_SCANNED,
                  HOSTS_CHANGED, PASSED_TOTAL, PASSED_CHANGED,
                  FAILED_TOTAL, FAILED_CHANGED, ERROR_TOTAL,
                  ERROR_CHANGED, COMPLIANCE)>
```

```
<!ELEMENT ASSET_GROUP
(#PCDATA|ASSET_GROUP_NAME|POLICY_TITLE|TECHNOLOGY|CONTROL_INSTANCES|HOSTS
_TOTAL|HOSTS_SCANNED|HOSTS_CHANGED|PASSED_TOTAL|PASSED_CHANGED|FAILED_TOT
AL|FAILED_CHANGED|ERROR_TOTAL|ERROR_CHANGED|COMPLIANCE)*>
<!ELEMENT ASSET_TAG (ASSET_TAG_NAME, POLICY_TITLE?, TECHNOLOGY?,
                     CONTROL_INSTANCES, HOSTS_TOTAL, HOSTS_SCANNED,
                     HOSTS_CHANGED, PASSED_TOTAL, PASSED_CHANGED,
                     FAILED_TOTAL, FAILED_CHANGED, ERROR_TOTAL,
                     ERROR_CHANGED, COMPLIANCE)>
<!ELEMENT CHANGED_TO_PASS (HOST*|CONTROL*|CRITICALITY*)>
<!ELEMENT CHANGED_TO_FAIL (HOST*|CONTROL*|CRITICALITY*)>
<!ELEMENT CHANGED_TO_ERROR (HOST*|CONTROL*|CRITICALITY*)>
<!ELEMENT HOST (IP_ADDRESS, NETBIOS, DNS, NETWORK?, ASSET_GROUP_NAME?,
                ASSET_TAG_NAME?, TECHNOLOGY, NUMBER_OF_POLICIES,
                PASSED_TOTAL?, PASSED_CHANGED?, FAILED_TOTAL?,
                FAILED_CHANGED?, ERROR_TOTAL?, ERROR_CHANGED?,
                COMPLIANCE, NETWORK?)>
<!ELEMENT CONTROL (ID, STATEMENT, COUNT)>
<!ELEMENT CRITICALITY
(#PCDATA|CRITICALITY_NAME|COUNT|PERCENT|ASSET_GROUP|ASSET_TAG|POLICY_TITL
E|TECHNOLOGY|CONTROL_INSTANCES|HOSTS_TOTAL|HOSTS_SCANNED|HOSTS_CHANGED|PA
SSED_TOTAL|PASSED_CHANGED|FAILED_TOTAL|FAILED_CHANGED|ERROR_TOTAL|ERROR_C
HANGED|COMPLIANCE|CONTROL_ID|STATEMENT)*>

<!ELEMENT OVERALL_COMPLIANCE_PERCENT (#PCDATA)>
<!ELEMENT UNIQUE_POLICES (#PCDATA)>
<!ELEMENT COUNT (#PCDATA)>
<!ELEMENT PERCENT (#PCDATA)>
<!ELEMENT TOTAL_CONTROL_DETECTED (#PCDATA)>
<!ELEMENT CHANGED_CONTROL (#PCDATA)>
<!ELEMENT TOTAL_HOSTS_IN_POLICIES (#PCDATA)>
<!ELEMENT SCANNED_HOSTS (#PCDATA)>
<!ELEMENT CHANGED (COUNT, PERCENT)>
<!ELEMENT TOTAL_TECHNOLOGY (#PCDATA)>
<!ELEMENT CHANGED_TECHNOLOGY (#PCDATA)>
<!ELEMENT NETWORK (#PCDATA)>


<!ELEMENT REPORT_TYPE (#PCDATA)>
<!ELEMENT GENERATION_DATETIME (#PCDATA)>

<!ELEMENT CREATED (#PCDATA)>
<!ELEMENT USER_NAME (#PCDATA)>
<!ELEMENT LOGIN_NAME (#PCDATA)>
<!ELEMENT USER_ROLE (#PCDATA)>

<!ELEMENT NAME (#PCDATA)>
<!ELEMENT ADDRESS (#PCDATA)>
<!ELEMENT CITY (#PCDATA)>
```

```
<!ELEMENT STATE (#PCDATA)>
<!ELEMENT COUNTRY (#PCDATA)>
<!ELEMENT ZIP_CODE (#PCDATA)>

<!ELEMENT TEMPLATE (#PCDATA)>
<!ELEMENT NUMBER_OF_POLICIES (#PCDATA)>
<!ELEMENT REPORT_TIMEFRAME (#PCDATA)>

<!ELEMENT INCLUDED_TAGS (#PCDATA)>
<!ELEMENT EXCLUDED_TAGS (#PCDATA)>

<!ELEMENT DETAIL_DATE (#PCDATA)>
<!ELEMENT POLICY_TITLE (#PCDATA)>
<!ELEMENT CONTROL_INSTANCES (#PCDATA)>
<!ELEMENT HOSTS_TOTAL (#PCDATA)>
<!ELEMENT HOSTS_SCANNED (#PCDATA)>
<!ELEMENT HOSTS_CHANGED (#PCDATA)>
<!ELEMENT PASSED_TOTAL (#PCDATA)>
<!ELEMENT PASSED_CHANGED (#PCDATA)>
<!ELEMENT FAILED_TOTAL (#PCDATA)>
<!ELEMENT FAILED_CHANGED (#PCDATA)>
<!ELEMENT ERROR_TOTAL (#PCDATA)>
<!ELEMENT ERROR_CHANGED (#PCDATA)>
<!ELEMENT COMPLIANCE (#PCDATA)>

<!ELEMENT POSTURE (#PCDATA)>
<!ELEMENT ASSET_GROUP_NAME (#PCDATA)>
<!ELEMENT ASSET_TAG_NAME (#PCDATA)>
<!ELEMENT IP_ADDRESS (#PCDATA)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT STATEMENT (#PCDATA)>
<!ELEMENT TOP (#PCDATA)>
<!ELEMENT NETBIOS (#PCDATA)>
<!ELEMENT DNS (#PCDATA)>
<!ELEMENT CRITICALITY_NAME (#PCDATA)>
<!ELEMENT TOTAL_FAILED_CONTROLS (#PCDATA)>
<!ELEMENT TOTAL_FAILED_CONTROLS_CHANGED (#PCDATA)>
<!ELEMENT CONTROL_ID (#PCDATA)>
```

XPaths for Compliance Scorecard Report

The XPaths for the compliance scorecard report are described below.

XPath	element specifications / notes
/COMPLIANCE_SCORECARD_REPORT	(ERROR | (HEADER, (SUMMARY) (DETAILS)))
/COMPLIANCE_SCORECARD_REPORT/ERROR (#PCDATA|COUNT|PERCENT)	An error message.
attribute: **number**	An error code, when available
/COMPLIANCE_SCORECARD_REPORT/HEADER (REPORT_TYPE, GENERATION_DATETIME)	
/COMPLIANCE_SCORECARD_REPORT/HEADER/REPORT_TYPE (#PCDATA)	The user defined report title.
/COMPLIANCE_SCORECARD_REPORT/HEADER/GENERATION_DATETIME (#PCDATA)	The date and time when the report was created.
COMPLIANCE_SCORECARD_REPORT/SUMMARY	(ABOUT_REPORT, REPORT_SETTINGS, REPORT_DISCOVERIES)
/COMPLIANCE_SCORECARD_REPORT/SUMMARY/ABOUT_REPORT	(REPORT_TYPE, CREATED, USER_NAME, LOGIN_NAME, USER_ROLE, COMPANY INFO)
/COMPLIANCE_SCORECARD_REPORT/SUMMARY/ABOUT_REPORT/REPORT_TYPE (#PCDATA)	Compliance scorecard report.
/COMPLIANCE_SCORECARD_REPORT/SUMMARY/ABOUT_REPORT/CREATED (#PCDATA)	The date and time the report was created.
/COMPLIANCE_SCORECARD_REPORT/SUMMARY/ABOUT_REPORT/USER_NAME (#PCDATA)	The name of the user who created the report.
/COMPLIANCE_SCORECARD_REPORT/SUMMARY/ABOUT_REPORT/LOGIN_NAME (#PCDATA)	The login ID of the user who created the report.
/COMPLIANCE_SCORECARD_REPORT/SUMMARY/ABOUT_REPORT/USER_ROLE (#PCDATA)	The user role assigned to the user who created the report: Manager, Unit Manager, Auditor, Scanner, or Reader.
./COMPLIANCE_SCORECARD_REPORT/SUMMARY/ABOUT_REPORT/COMPANY INFO)	(NAME, ADDRESS, CITY, STATE, COUNTRY, ZIP_CODE)
	The user's company name and address, as defined in the user's account.
/COMPLIANCE_SCORECARD_REPORT/SUMMARY/REPORT_SETTINGS	(TEMPLATE, NUMBER_OF_POLICIES, REPORT_TIMEFRAME, ASSET_GROUPS*, ASSET_TAGS*, CRITICALITY*)
/COMPLIANCE_SCORECARD_REPORT/SUMMARY/REPORT_SETTINGS/TEMPLATE (#PCDATA)	The name of the template used to generate the report.

XPath	element specifications / notes
/COMPLIANCE_SCORECARD_REPORT/SUMMARY/REPORT_SETTINGS/NUMBER_OF_POLICIES (#PCDATA)	
	The number of policies selected for the report.
/COMPLIANCE_SCORECARD_REPORT/SUMMARY/REPORT_SETTINGS/REPORT_TIMEFRAME (#PCDATA)	
	The date range reported on.
/COMPLIANCE_SCORECARD_REPORT/SUMMARY/REPORT_SETTINGS/ASSET_GROUPS	
ASSET_GROUP_NAME (#PCDATA)	
	An asset group name.
/COMPLIANCE_SCORECARD_REPORT/SUMMARY/REPORT_SETTINGS/ASSET_TAGS	
	((INCLUDED_TAGS, EXCLUDED_TAGS?) | ASSET_TAG?)
	The asset tags selected for the report.
/COMPLIANCE_SCORECARD_REPORT/SUMMARY/REPORT_SETTINGS/CRITICALITY (#PCDATA)	
	The criticality levels included in the report.
/COMPLIANCE_SCORECARD_REPORT/SUMMARY/REPORT_DISCOVERIES	
	(OVERALL_COMPLIANCE, BY_CONTROL, BY_HOSTS, BY_TECHNOLOGY, BY_CRITICALITY*)
/COMPLIANCE_SCORECARD_REPORT/SUMMARY/REPORT_DISCOVERIES/OVERALL_COMPLIANCE	
	(OVERALL_COMPLIANCE_PERCENT, UNIQUE_POLICES, PASSED, FAILED, ERROR)
/COMPLIANCE_SCORECARD_REPORT/SUMMARY/REPORT_DISCOVERIES/OVERALL_COMPLIANCE/ OVERALL_COMPLIANCE_PERCENT (#PCDATA)	
	The percent of compliance across all policies included in the report.
/COMPLIANCE_SCORECARD_REPORT/SUMMARY/REPORT_DISCOVERIES/OVERALL_COMPLIANCE/ UNIQUE_POLICES (#PCDATA)	
	The number of unique policies included in the report.
/COMPLIANCE_SCORECARD_REPORT/SUMMARY/REPORT_DISCOVERIES/OVERALL_COMPLIANCE/ PASSED (COUNT, PERCENT)	
	The number and percent of controls that passed.
/COMPLIANCE_SCORECARD_REPORT/SUMMARY/REPORT_DISCOVERIES/OVERALL_COMPLIANCE/ FAILED (COUNT, PERCENT)	
	The number and percent of controls that failed.
/COMPLIANCE_SCORECARD_REPORT/SUMMARY/REPORT_DISCOVERIES/OVERALL_COMPLIANCE/ ERROR (COUNT, PERCENT)	
	The number and percent of controls with an Error status in the report. An error status is returned for a custom control if an error occurred during control evaluation (and the ignore errors configuration option was not selected).
/COMPLIANCE_SCORECARD_REPORT/SUMMARY/REPORT_DISCOVERIES/BY_CONTROL	
	(TOTAL_CONTROL_DETECTED, CHANGED_CONTROL, PASSED, FAILED, ERROR)
/COMPLIANCE_SCORECARD_REPORT/SUMMARY/REPORT_DISCOVERIES/BY_CONTROL/ TOTAL_CONTROL_DETECTED (#PCDATA)	

XPath	element specifications / notes
	The number of controls detected.
/COMPLIANCE_SCORECARD_REPORT/SUMMARY/REPORT_DISCOVERIES/BY_CONTROL/CHANGED_CONTROL (#PCDATA)	
	The number of changed controls detected.
/COMPLIANCE_SCORECARD_REPORT/SUMMARY/REPORT_DISCOVERIES/BY_CONTROL/PASSED	
	(COUNT, PERCENT)
	The number and percent of controls passed.
/COMPLIANCE_SCORECARD_REPORT/SUMMARY/REPORT_DISCOVERIES/BY_CONTROL/FAILED	
	(COUNT, PERCENT) (#PCDATA)
	The number and percent of controls failed.
/COMPLIANCE_SCORECARD_REPORT/SUMMARY/REPORT_DISCOVERIES/BY_CONTROL/ERROR	
	(COUNT, PERCENT) (#PCDATA)
	The number and percent of controls in error.
/COMPLIANCE_SCORECARD_REPORT/SUMMARY/REPORT_DISCOVERIES/BY_HOSTS	
	(TOTAL_HOSTS_IN_POLICIES, SCANNED_HOSTS, CHANGED)
/COMPLIANCE_SCORECARD_REPORT/SUMMARY/REPORT_DISCOVERIES/BY_HOSTS/TOTAL_HOSTS_IN_POLICIES (#PCDATA)	
	The number of hosts in the selected policies.
/COMPLIANCE_SCORECARD_REPORT/SUMMARY/REPORT_DISCOVERIES/BY_HOSTS/SCANNED_HOSTS (#PCDATA)	
	The number of scanned hosts included in the selected policies.
/COMPLIANCE_SCORECARD_REPORT/SUMMARY/REPORT_DISCOVERIES/BY_HOSTS/CHANGED	
	(COUNT, PERCENT)
	The number and percent changed hosts
/COMPLIANCE_SCORECARD_REPORT/SUMMARY/REPORT_DISCOVERIES/BY_TECHNOLOGY	
	((TOTAL_TECHNOLOGY, CHANGED_TECHNOLOGY, TECHNOLOGY*)|(TECHNOLOGY+))
/COMPLIANCE_SCORECARD_REPORT/SUMMARY/REPORT_DISCOVERIES/BY_TECHNOLOGY/TOTAL_TECHNOLOGY (#PCDATA)	
	The number of technologies included in the report.
/COMPLIANCE_SCORECARD_REPORT/SUMMARY/REPORT_DISCOVERIES/BY_TECHNOLOGY/CHANGED_TECHNOLOGY (#PCDATA)	
	The number of changed technologies in the report.
/COMPLIANCE_SCORECARD_REPORT/SUMMARY/REPORT_DISCOVERIES/BY_TECHNOLOGY/TECHNOLOGY* (NAME, COUNT, PERCENT)	
	The technology name, count and percent.
/COMPLIANCE_SCORECARD_REPORT/SUMMARY/REPORT_DISCOVERIES/BY_CRITICALITY	
	(TOTAL_FAILED_CONTROLS*,TOTAL_FAILED_CONTROLS_CHANGED*, CRITICALITY*)

XPath	element specifications / notes
/COMPLIANCE_SCORECARD_REPORT/SUMMARY/REPORT_DISCOVERIES/BY_CRITICALITY/TOTAL_FAILED_CONTROLS* (#PCDATA)	
	The number of failed controls in the report.
/COMPLIANCE_SCORECARD_REPORT/SUMMARY/REPORT_DISCOVERIES/BY_CRITICALITY/TOTAL_FAILED_CONTROLS_CHANGED* (#PCDATA)	
	The number of controls that changed to fail in the report time frame.
/COMPLIANCE_SCORECARD_REPORT/SUMMARY/REPORT_DISCOVERIES/BY_CRITICALITY/CRITICALITY* (NAME, COUNT, PERCENT)	
	The number and percentage of controls that changed to fail for each criticality.
/COMPLIANCE_SCORECARD_REPORT/DETAILS	
	(COMPLIANCE_BY_POLICY*, COMPLIANCE_BY_ASSET_GROUP*, COMPLIANCE_BY_ASSET_TAG*, COMPLIANCE_BY_TECHNOLOGY*, COMPLIANCE_BY_CRITICALITY*, TOP_HOST_WITH_CHANGES*, TOP_CONTROLS_WITH_CHANGES*, FAILED_CONTROLS_BY_CRITICALITY*)

SCAP Policy List Output

The SCAP policy list XML is returned from an API request for an SCAP policy list. The DTD can be found at the following URL (where qualysapi.qualys.com is the API server URL where your account is located:

```
https://qualysapi.qualys.com/api/2.0/fo/compliance/fdcc_policy/fdcc_policy_list_output.dtd
```

DTD for SCAP Policy List Output

A recent DTD for the SCAP policy list is shown below.

```
<!-- QUALYS FDCC_POLICY_LIST_OUTPUT DTD -->
<!ELEMENT FDCC_POLICY_LIST_OUTPUT (REQUEST?,RESPONSE)>

<!ELEMENT REQUEST (DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?,
                   POST_DATA?)>
<!ELEMENT DATETIME (#PCDATA)>
<!ELEMENT USER_LOGIN (#PCDATA)>
<!ELEMENT RESOURCE (#PCDATA)>
<!ELEMENT PARAM_LIST (PARAM+)>
<!ELEMENT PARAM (KEY, VALUE)>
<!ELEMENT KEY (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>
<!-- if returned, POST_DATA will be urlencoded -->
<!ELEMENT POST_DATA (#PCDATA)>

<!ELEMENT RESPONSE (DATETIME, (FDCC_POLICY_LIST|ID_SET)?, WARNING_LIST?)>
<!ELEMENT FDCC_POLICY_LIST (FDCC_POLICY+)>
<!ELEMENT FDCC_POLICY (ID, TITLE, DESCRIPTION, BENCHMARK,
                       BENCHMARK_PROFILE, BENCHMARK_STATUS_DATE, VERSION,
                       TECHNOLOGY, NIST_PROVIDED, CREATED, LAST_MODIFIED,
                       ASSET_GROUP_LIST?, FDCC_FILE_LIST?)>

<!ELEMENT ID (#PCDATA)>
<!ELEMENT TITLE (#PCDATA)>
<!ELEMENT DESCRIPTION (#PCDATA)>
<!ELEMENT BENCHMARK (#PCDATA)>
<!ELEMENT BENCHMARK_PROFILE (#PCDATA)>
<!ELEMENT BENCHMARK_STATUS_DATE (#PCDATA)>
<!ELEMENT VERSION (#PCDATA)>
<!ELEMENT TECHNOLOGY (#PCDATA)>
<!ELEMENT NIST_PROVIDED (#PCDATA)>

<!ELEMENT CREATED (DATETIME, BY)>
<!ELEMENT BY (#PCDATA)>
```

```
<!ELEMENT LAST_MODIFIED (DATETIME, BY)>

<!ELEMENT ASSET_GROUP_LIST (ASSET_GROUP+)>
<!ELEMENT ASSET_GROUP (ID, TITLE)>

<!ELEMENT FDCC_FILE_LIST (FDCC_FILE+)>
<!ELEMENT FDCC_FILE (FILE_NAME, FILE_HASH)>
<!ELEMENT FILE_NAME (#PCDATA)>
<!ELEMENT FILE_HASH (#PCDATA)>

<!ELEMENT WARNING_LIST (WARNING+)>
<!ELEMENT WARNING (CODE?, TEXT, URL?)>
<!ELEMENT CODE (#PCDATA)>
<!ELEMENT TEXT (#PCDATA)>
<!ELEMENT URL (#PCDATA)>

<!-- EOF -->
```

XPaths for SCAP Policy List Output

This section describes the XPaths for the SCAP policy list output.

SCAP Policy List Output: Request

XPath	element specifications / notes
/FDCC_POLICY_LIST_OUTPUT	(REQUEST?, RESPONSE)
/FDCC_POLICY_LIST_OUTPUT/REQUEST	(DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?, POST_DATA?)
/FDCC_POLICY_LIST_OUTPUT/REQUEST/DATETIME	(#PCDATA) The date and time of the request.
/FDCC_POLICY_LIST_OUTPUT/REQUEST/USER_LOGIN	(#PCDATA) The user login ID of the user who made the request.
/FDCC_POLICY_LIST_OUTPUT/REQUEST/RESOURCE	(#PCDATA) The resource specified for the request.
/FDCC_POLICY_LIST_OUTPUT/REQUEST/PARAM_LIST	(PARAM+)
/FDCC_POLICY_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM	(KEY, VALUE)
/FDCC_POLICY_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM/KEY	(#PCDATA) An input parameter name.
/FDCC_POLICY_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM/VALUE	(#PCDATA) An input parameter value.
/FDCC_POLICY_LIST_OUTPUT/REQUEST/POST_DATA	(#PCDATA) The POST data, if any.

SCAP Policy List Output: Response

XPath	element specifications / notes
/FDCC_POLICY_LIST_OUTPUT	(REQUEST?, RESPONSE)
/FDCC_POLICY_LIST_OUTPUT/RESPONSE	(DATETIME, (FDCC_POLICY_LIST|ID_SET)?, WARNING_LIST?)
/FDCC_POLICY_LIST_OUTPUT/RESPONSE/DATETIME	(#PCDATA) The date and time of the response.
/FDCC_POLICY_LIST_OUTPUT/RESPONSE/FDCC_POLICY_LIST	(FDCC_POLICY+)
/FDCC_POLICY_LIST_OUTPUT/RESPONSE/POLICY_LIST/FDCC_POLICY	(ID, TITLE, DESCRIPTION, BENCHMARK, BENCHMARK_PROFILE, BENCHMARK_STATUS_DATE, VERSION, TECHNOLOGY, NIST_PROVIDED, CREATED, LAST_MODIFIED, ASSET_GROUP_LIST?, FDCC_FILE_LIST?)
/FDCC_POLICY_LIST_OUTPUT/RESPONSE/FDCC_POLICY_LIST/FDCC_POLICY/ID	(#PCDATA) A SCAP policy ID.
/FDCC_POLICY_LIST_OUTPUT/RESPONSE/FDCC_POLICY_LIST/FDCC_POLICY/TITLE	(#PCDATA) A SCAP policy title.
/FDCC_POLICY_LIST_OUTPUT/RESPONSE/FDCC_POLICY_LIST/FDCC_POLICY/DESCRIPTION	(#PCDATA) A description of the SCAP policy.
/FDCC_POLICY_LIST_OUTPUT/RESPONSE/FDCC_POLICY_LIST/FDCC_POLICY/BENCHMARK	(#PCDATA) The SCAP benchmark defined for the FDCC policy.
/FDCC_POLICY_LIST_OUTPUT/RESPONSE/FDCC_POLICY_LIST/FDCC_POLICY/BENCHMARK_PROFILE	(#PCDATA) The SCAP profile that is defined for the FDCC policy in the FDCC Content.
/FDCC_POLICY_LIST_OUTPUT/RESPONSE/FDCC_POLICY_LIST/FDCC_POLICY/BENCHMARK_STATUS_DATE	(#PCDATA) The SCAP status date, as defined for the FDCC policy in the SCAP XCCDF file.
/FDCC_POLICY_LIST_OUTPUT/RESPONSE/FDCC_POLICY_LIST/FDCC_POLICY/VERSION	(#PCDATA) The base version of the SCAP policy as defined by NIST, when the policy is a NIST provided policy.
/FDCC_POLICY_LIST_OUTPUT/RESPONSE/FDCC_POLICY_LIST/FDCC_POLICY/TECHNOLOGY	(#PCDATA) The technology defined for the SCAP policy.
/FDCC_POLICY_LIST_OUTPUT/RESPONSE/FDCC_POLICY_LIST/FDCC_POLICY/NIST_PROVIDED	(#PCDATA) Yes indicates the SCAP policy was provided by NIST. No indicates the SCAP policy is a user-defined custom policy.

XPath	element specifications / notes
/FDCC_POLICY_LIST_OUTPUT/RESPONSE/FDCC_POLICY_LIST/FDCC_POLICY/CREATED (DATETIME, BY)	
/FDCC_POLICY_LIST_OUTPUT/RESPONSE/FDCC_POLICY_LIST/FDCC_POLICY/CREATED/DATETIME (#PCDATA)	
	The date/time when the SCAP policy was first created.
/FDCC_POLICY_LIST_OUTPUT/RESPONSE/FDCC_POLICY_LIST/FDCC_POLICY/CREATED/BY (#PCDATA)	
	The user login ID of the user who first created the SCAP policy.
/FDCC_POLICY_LIST_OUTPUT/RESPONSE/FDCC_POLICY_LIST/FDCC_POLICY/LAST_MODIFIED (DATETIME, BY)	
/FDCC_POLICY_LIST_OUTPUT/RESPONSE/FDCC_POLICY_LIST/FDCC_POLICY/LAST_MODIFIED/DATETIME (#PCDATA)	
	The date/time when the policy was last updated.
/FDCC_POLICY_LIST_OUTPUT/RESPONSE/FDCC_POLICY_LIST/FDCC_POLICY/LAST_MODIFIED/BY (#PCDATA)	
	The user login ID of the user who last modified the policy.
/FDCC_POLICY_LIST_OUTPUT/RESPONSE/FDCC_POLICY_LIST/FDCC_POLICY/ASSET_GROUP_LIST (ASSET_GROUP+)	
/FDCC_POLICY_LIST_OUTPUT/RESPONSE/FDCC_POLICY_LIST/FDCC_POLICY/ASSET_GROUP_LIST/ASSET_GROUP (ID, TITLE)	
/FDCC_POLICY_LIST_OUTPUT/RESPONSE/FDCC_POLICY_LIST/FDCC_POLICY/ASSET_GROUP_LIST/ASSET_GROUP/ID (#PCDATA)	
	The ID of an asset group assigned to the SCAP policy.
/FDCC_POLICY_LIST_OUTPUT/RESPONSE/FDCC_POLICY_LIST/FDCC_POLICY/ASSET_GROUP_LIST/ASSET_GROUP/TITLE (#PCDATA)	
	The title of an asset group assigned to the SCAP policy.
/FDCC_POLICY_LIST_OUTPUT/RESPONSE/FDCC_POLICY_LIST/FDCC_POLICY/FDCC_FILE_LIST (FDCC_FILE+)	
/FDCC_POLICY_LIST_OUTPUT/RESPONSE/FDCC_POLICY_LIST/FDCC_POLICY/FDCC_FILE_LIST/FDCC_FILE (FILE_NAME, FILE_HASH	
/FDCC_POLICY_LIST_OUTPUT/RESPONSE/FDCC_POLICY_LIST/FDCC_POLICY/FDCC_FILE_LIST/FDCC_FILE/FILE_NAME (#PCDATA)	
	A SCAP file name.
/FDCC_POLICY_LIST_OUTPUT/RESPONSE/FDCC_POLICY_LIST/FDCC_POLICY/FDCC_FILE_LIST/FDCC_FILE/FILE_HASH (#PCDATA)	
	The MD5 hash of a SCAP file name.
/FDCC_POLICY_LIST_OUTPUT/RESPONSE/ID_SET (ID|ID_RANGE)	
/FDCC_POLICY_LIST_OUTPUT/RESPONSE/ID_SET/ID (#PCDATA)	
	A SCAP policy ID.
/FDCC_POLICY_LIST_OUTPUT/RESPONSE/ID_SET/ID_RANGE (#PCDATA)	
	A range SCAP policy IDs.

SCAP Policy List Output: Warning

XPath	element specifications / notes
/FDCC_POLICY_LIST_OUTPUT/RESPONSE/WARNING_LIST (WARNING+)	
/FDCC_POLICY_LIST_OUTPUT/RESPONSE/WARNING_LIST/WARNING (CODE?, TEXT, URL?)	
/FDCC_POLICY_LIST_OUTPUT/RESPONSE/WARNING/CODE (#PCDATA)	
	A warning code. A warning code appears when the API request identifies more than 1,000 records (policies).
/FDCC_POLICY_LIST_OUTPUT/RESPONSE/WARNING/TEXT (#PCDATA)	
	A warning message. A warning message appears when the API request identifies more than 1,000 records (policies).
/FDCC_POLICY_LIST_OUTPUT/RESPONSE/WARNING/URL (#PCDATA)	
	The URL for making another API request for the next batch of SCAP policy records.

APPENDIX D

Scan Authentication XML

This appendix describes the XML output returned from API V2 requests using the Scan Authentication API functions.

- Authentication Record List Output
- Authentication Record List by Type Output
- Authentication Record Return
- Authentication Vault List Output
- Authentication Vault View Output

Authentication Record List Output

The authentication record list XML is returned from an authentication record list (**/auth**) API call.

The DTD can be found at the following URL:

`https://<qualysapi.qualys.com>/api/2.0/fo/auth/auth_records.dtd`

where `<qualysapi.qualys.com>` is the API server URL where your account is located.

DTD for Authentication Record List Output

A recent DTD for the authentication record list is shown below.

```
<!-- QUALYS AUTH_RECORDS_OUTPUT DTD -->

<!ELEMENT AUTH_RECORDS_OUTPUT (REQUEST?, RESPONSE)>

<!ELEMENT REQUEST (DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?,
                   POST_DATA?)>
<!ELEMENT DATETIME (#PCDATA)>
<!ELEMENT USER_LOGIN (#PCDATA)>
<!ELEMENT RESOURCE (#PCDATA)>
<!ELEMENT PARAM_LIST (PARAM+)>
<!ELEMENT PARAM (KEY, VALUE)>
<!ELEMENT KEY (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>
<!-- if returned, POST_DATA will be urlencoded -->
<!ELEMENT POST_DATA (#PCDATA)>

<!ELEMENT RESPONSE (DATETIME, AUTH_RECORDS?, WARNING_LIST?)>
<!ELEMENT AUTH_RECORDS (AUTH_UNIX_IDS?, AUTH_WINDOWS_IDS?,
                        AUTH_ORACLE_IDS?, AUTH_ORACLE_LISTENER_IDS?,
                        AUTH_SNMP_IDS?, AUTH_MS_SQL_IDS?,
                        AUTH_IBM_DB2_IDS?, AUTH_VMWARE_IDS?,
                        AUTH_MS_IIS_IDS?, AUTH_APACHE_IDS?,
                        AUTH_IBM_WEBSPHERE_IDS?, AUTH_HTTP_IDS?,
                        AUTH_MYSQL_IDS?, AUTH_TOMCAT_IDS?)>

<!ELEMENT AUTH_UNIX_IDS (ID_SET)>
<!ELEMENT AUTH_WINDOWS_IDS (ID_SET)>
<!ELEMENT AUTH_ORACLE_IDS (ID_SET)>
<!ELEMENT AUTH_ORACLE_LISTENER_IDS (ID_SET)>
<!ELEMENT AUTH_SNMP_IDS (ID_SET)>
<!ELEMENT AUTH_MS_SQL_IDS (ID_SET)>
<!ELEMENT AUTH_IBM_DB2_IDS (ID_SET)>
```

```
<!ELEMENT AUTH_VMWARE_IDS (ID_SET)>
<!ELEMENT AUTH_MS_IIS_IDS (ID_SET)>
<!ELEMENT AUTH_APACHE_IDS (ID_SET)>
<!ELEMENT AUTH_IBM_WEBSPHERE_IDS (ID_SET)>
<!ELEMENT AUTH_HTTP_IDS (ID_SET)>
<!ELEMENT AUTH_MYSQL_IDS (ID_SET)>
<!ELEMENT AUTH_TOMCAT_IDS (ID_SET)>

<!ELEMENT WARNING_LIST (WARNING+)>
<!ELEMENT WARNING (CODE?, TEXT, URL?, ID_SET?)>
<!ELEMENT CODE (#PCDATA)>
<!ELEMENT TEXT (#PCDATA)>
<!ELEMENT URL (#PCDATA)>

<!ELEMENT ID_SET (ID|ID_RANGE)+>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT ID_RANGE (#PCDATA)>

<!-- EOF -->
```

XPaths for Authentication Record List Output

This section describes the XPaths for the authentication record list output.

Authentication Record List Output: Request

XPath	element specifications / notes
/AUTH_RECORDS_OUTPUT	(REQUEST?, RESPONSE)
/AUTH_RECORDS_OUTPUT/REQUEST	(DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?, POST_DATA?)
/AUTH_RECORDS_OUTPUT/REQUEST/DATETIME	(#PCDATA) The date and time of the API request.
/AUTH_RECORDS_OUTPUT/REQUEST/USER_LOGIN	(#PCDATA) The user login ID of the user who made the request.
/AUTH_RECORDS_OUTPUT/REQUEST/RESOURCE	(#PCDATA) The resource specified for the request.
/AUTH_RECORDS_OUTPUT/REQUEST/PARAM_LIST	(PARAM+)
/AUTH_RECORDS_OUTPUT/REQUEST/PARAM_LIST/PARAM	(KEY, VALUE)
/AUTH_RECORDS_OUTPUT/REQUEST/PARAM_LIST/PARAM/KEY	(#PCDATA) An input parameter name.
/AUTH_RECORDS_OUTPUT/REQUEST/PARAM_LIST/PARAM/VALUE	(#PCDATA) An input parameter value.
/AUTH_RECORDS_OUTPUT/REQUEST/POST_DATA	(#PCDATA) The POST data, if any.

Authentication Record List Output: Response

XPath	element specifications / notes
/AUTH_RECORDS_OUTPUT	(REQUEST?, RESPONSE)
/AUTH_RECORDS_OUTPUT/RESPONSE	(DATETIME, AUTH_RECORDS?, WARNING_LIST?)
/AUTH_RECORDS_OUTPUT/RESPONSE/DATETIME	(#PCDATA) The date and time of the response.
/AUTH_RECORDS_OUTPUT/RESPONSE/AUTH_RECORDS	(AUTH_UNIX_IDS?, AUTH_WINDOWS_IDS?, AUTH_ORACLE_IDS?, AUTH_ORACLE_LISTENER_IDS?, AUTH_SNMP_IDS?, AUTH_MS_SQL_IDS?, AUTH_IBM_DB2_IDS?, AUTH_VMWARE_IDS?, AUTH_MS_IIS_IDS?, AUTH_APACHE_IDS?, AUTH_IBM_WEBSPHERE_IDS?, AUTH_HTTP_IDS?, AUTH_MYSQL_IDS?, AUTH_TOMCAT_IDS?)
/AUTH_RECORDS_OUTPUT/RESPONSE/AUTH_RECORDS/AUTH_UNIX_IDS	(ID_SET) A set of Unix and Cisco IOS authentication record IDs.
/AUTH_RECORDS_OUTPUT/RESPONSE/AUTH_RECORDS/AUTH_WINDOWS_IDS	(ID_SET) A set of Windows authentication record IDs.
/AUTH_RECORDS_OUTPUT/RESPONSE/AUTH_RECORDS/AUTH_ORACLE_IDS	(ID_SET) A set of Oracle authentication record IDs.
/AUTH_RECORDS_OUTPUT/RESPONSE/AUTH_RECORDS/AUTH_ORACLE_LISTENER_IDS	(ID_SET) A set of Oracle Listener authentication record IDs.
/AUTH_RECORDS_OUTPUT/RESPONSE/AUTH_RECORDS/AUTH_SNMP_IDS	(ID_SET) A set of SNMP authentication record IDs.
/AUTH_RECORDS_OUTPUT/RESPONSE/AUTH_RECORDS/AUTH_MS_SQL_IDS	(ID_SET) A set of MS SQL authentication record IDs.
/AUTH_RECORDS_OUTPUT/RESPONSE/AUTH_RECORDS/AUTH_IBM_DB2_IDS	(ID_SET) A set of IBM DB2 authentication record IDs.
/AUTH_RECORDS_OUTPUT/RESPONSE/AUTH_RECORDS/AUTH_VMWARE_IDS	(ID_SET) A set of VMware authentication record IDs.
/AUTH_RECORDS_OUTPUT/RESPONSE/AUTH_RECORDS/AUTH_AUTH_MS_IIS_IDS	(ID_SET) A set of Microsoft IIS Web Server authentication record IDs.
/AUTH_RECORDS_OUTPUT/RESPONSE/AUTH_RECORDS/AUTH_APACHE_IDS?	(ID_SET) A set of Apache Web Server authentication record IDs.
/AUTH_RECORDS_OUTPUT/RESPONSE/AUTH_RECORDS/AUTH_IBM_WEBSPHERE_IDS	(ID_SET) A set of IBM WebSphere Application Server authentication record IDs.
/AUTH_RECORDS_OUTPUT/RESPONSE/AUTH_RECORDS/AUTH_HTTP_IDS	(ID_SET) A set of HTTP authentication record IDs.
/AUTH_RECORDS_OUTPUT/RESPONSE/AUTH_RECORDS/AUTH_MYSQL_IDS	(ID_SET) A set of MySQL authentication record IDs.

XPath	element specifications / notes
/AUTH_RECORDS_OUTPUT/RESPONSE/AUTH_RECORDS/AUTH_TOMCAT_IDS (ID_SET)	
	A set of Tomcat Server authentication record IDs.

Authentication Record List Output: Warning List

XPath	element specifications / notes
/AUTH_RECORDS_OUTPUT/RESPONSE/WARNING_LIST (WARNING+)	
/AUTH_RECORDS_OUTPUT/RESPONSE/WARNING_LIST/WARNING (CODE?, TEXT, URL?, ID_SET?)	
/AUTH_RECORDS_OUTPUT/RESPONSE/WARNING_LIST/WARNING/CODE (#PCDATA)	
	A warning code. A warning code appears when the API request identifies more than 1,000 records.
/AUTH_RECORDS_OUTPUT/RESPONSE/WARNING_LIST/WARNING/TEXT (#PCDATA)	
	A warning message. A warning message appears when the API request identifies more than 1,000 records.
/AUTH_RECORDS_OUTPUT/RESPONSE/WARNING_LIST/WARNING/URL (#PCDATA)	
	The URL for making another API request for the next batch of authentication records.
/AUTH_RECORDS_OUTPUT/RESPONSE/WARNING_LIST/WARNING/ID_SET (ID|ID_RANGE)	
/AUTH_RECORDS_OUTPUT/RESPONSE/WARNING_LIST/WARNING/ID_SET/ID (#PCDATA)	
	An authentication record ID.
/AUTH_RECORDS_OUTPUT/RESPONSE/WARNING_LIST/WARNING/ID_SET/ID_RANGE (#PCDATA)	
	A range of authentication record IDs.

Authentication Record List by Type Output

The authentication record list by type XML is returned from an API request (**/auth/<type>**) for an authentication record list by type.

The DTD can be found at the following URL:

```
https://<qualysapi.qualys.com>/api/2.0/fo/auth/<type>/
auth_<type>_list_output.dtd
```

where <qualysapi.qualys.com> is the API server URL where your account is located and <type> is an authentication type: unix, windows, oracle, oracle_listener, snmp, ms_sql, mysql, vmware, ibm_db2, apache, ms_iis, ibm_websphere, tomcat, http.

DTD for Authentication Record List by Type Output

A recent DTD for the authentication record by type list is shown below. The following example is for the Windows authentication type. The response output will be different for different authentication types.

```
<!-- QUALYS AUTH_WINDOWS_LIST_OUTPUT DTD -->
<!ELEMENT AUTH_WINDOWS_LIST_OUTPUT (REQUEST?, RESPONSE)>

<!ELEMENT REQUEST (DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?,
POST_DATA?)>
<!ELEMENT DATETIME (#PCDATA)>
<!ELEMENT USER_LOGIN (#PCDATA)>
<!ELEMENT RESOURCE (#PCDATA)>
<!ELEMENT PARAM_LIST (PARAM+)>
<!ELEMENT PARAM (KEY, VALUE)>
<!ELEMENT KEY (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>
<!-- if returned, POST_DATA will be urlencoded -->
<!ELEMENT POST_DATA (#PCDATA)>

<!ELEMENT RESPONSE (DATETIME, (AUTH_WINDOWS_LIST|ID_SET)?, WARNING_LIST?,
                    GLOSSARY?)>
<!ELEMENT AUTH_WINDOWS_LIST (AUTH_WINDOWS+)>

<!-- If WINDOWS_DOMAIN is set, then IP_SET is optional (not specified
means service selects IPs) -->
<!ELEMENT AUTH_WINDOWS (ID, TITLE, USERNAME, NTLM?, WINDOWS_DOMAIN?,
                        WINDOWS_AD_DOMAIN?, WINDOWS_AD_TRUST?, IP_SET?,
                        NETWORK_ID?, CREATED, LAST_MODIFIED, COMMENTS?,
                        USE_AGENTLESS_TRACKING?)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT TITLE (#PCDATA)>
<!ELEMENT USERNAME (#PCDATA)>
```

```
<!ELEMENT NTLM (#PCDATA)>
<!ELEMENT WINDOWS_DOMAIN (#PCDATA)>
<!ELEMENT WINDOWS_AD_DOMAIN (#PCDATA)>
<!ELEMENT WINDOWS_AD_TRUST (#PCDATA)>

<!ELEMENT IP_SET (IP|IP_RANGE)+>
<!ELEMENT IP (#PCDATA)>
<!ELEMENT IP_RANGE (#PCDATA)>
<!ELEMENT NETWORK_ID (#PCDATA)>
<!ELEMENT CREATED (DATETIME, BY)>
<!ELEMENT BY (#PCDATA)>
<!ELEMENT LAST_MODIFIED (DATETIME)>
<!ELEMENT COMMENTS (#PCDATA)>
<!ELEMENT USE_AGENTLESS_TRACKING (#PCDATA)>

<!ELEMENT WARNING_LIST (WARNING+)>
<!ELEMENT WARNING (CODE?, TEXT, URL?, ID_SET?)>
<!ELEMENT CODE (#PCDATA)>
<!ELEMENT TEXT (#PCDATA)>
<!ELEMENT URL (#PCDATA)>
<!ELEMENT ID_SET (ID|ID_RANGE)+>
<!ELEMENT ID_RANGE (#PCDATA)>

<!ELEMENT GLOSSARY (USER_LIST?)>
<!ELEMENT USER_LIST (USER+)>
<!ELEMENT USER (USER_LOGIN, FIRST_NAME, LAST_NAME)>
<!ELEMENT FIRST_NAME (#PCDATA)>
<!ELEMENT LAST_NAME (#PCDATA)>

<!-- EOF -->
```

XPaths for Authentication Record List by Type Output

This section describes the XPaths for the authentication record list by type output.

Authentication Record List by Type Output: Request

Note that <TYPE> is the authentication type, such as unix, windows, oracle, snmp, ms_sql, ibm_db2.

XPath	element specifications / notes
/AUTH_<TYPE>_LIST_OUTPUT	(REQUEST?, RESPONSE)
/AUTH_<TYPE>_LIST_OUTPUT/REQUEST	(DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?, POST_DATA?)
/AUTH_<TYPE>_LIST_OUTPUT/REQUEST/DATETIME	(#PCDATA) The date and time of the API request.

XPath	element specifications / notes
/AUTH_<TYPE>_LIST_OUTPUT/REQUEST/USER_LOGIN (#PCDATA)	
	The user login ID of the user who made the request.
/AUTH_<TYPE>_LIST_OUTPUT/REQUEST/RESOURCE (#PCDATA)	
	The resource specified for the request.
/AUTH_<TYPE>_LIST_OUTPUT/REQUEST/PARAM_LIST (PARAM+)	
/AUTH_<TYPE>_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM (KEY, VALUE)	
/AUTH_<TYPE>_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM/KEY (#PCDATA)	
	An input parameter name.
/AUTH_<TYPE>_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM/VALUE (#PCDATA)	
	An input parameter value.
/AUTH_<TYPE>_LIST_OUTPUT/REQUEST/POST_DATA (#PCDATA)	
	The POST data, if any. POST data is urlencoded.

Authentication Record List by Type Output: Response

Note that <TYPE> is the authentication type, such as unix, windows, oracle, snmp, etc. Elements that appear in the response for all types are described below. Type-specific elements are described in the sections that follow.

XPath	element specifications / notes
/AUTH_<TYPE>_LIST_OUTPUT (REQUEST?, RESPONSE)	
/AUTH_<TYPE>_LIST_OUTPUT/RESPONSE	(DATETIME, (AUTH_<TYPE>_LIST|ID_SET)?, WARNING_LIST? GLOSSARY?)
/AUTH_<TYPE>_LIST_OUTPUT/RESPONSE/DATETIME (#PCDATA)	
	The date and time of the response.
/AUTH_<TYPE>_LIST_OUTPUT/RESPONSE/AUTH_<TYPE>_LIST (AUTH_<TYPE>+)	
/AUTH_<TYPE>_LIST_OUTPUT/RESPONSE/AUTH_<TYPE>_LIST/AUTH_<TYPE>	(**ID, TITLE,** <type-specific elements>,**IP_SET?, NETWORK_ID?, CREATED, LAST_MODIFIED, COMMENTS?**)
/AUTH_<TYPE>_LIST_OUTPUT/RESPONSE/AUTH_<TYPE>_LIST/AUTH_<TYPE>/ID (#PCDATA)	
	The authentication record ID.
/AUTH_<TYPE>_LIST_OUTPUT/RESPONSE/AUTH_<TYPE>_LIST/AUTH_<TYPE>/TITLE (#PCDATA)	
	The authentication record title.
/AUTH_<TYPE>_LIST_OUTPUT/RESPONSE/AUTH_<TYPE>_LIST/AUTH_<TYPE>/IP_SET (IP|IP_RANGE)	
/AUTH_<TYPE>_LIST_OUTPUT/RESPONSE/AUTH_<TYPE>_LIST/AUTH_<TYPE>/IP_SET/IP (#PCDATA)	
	An IP address saved in the authentication record.
/AUTH_<TYPE>_LIST_OUTPUT/RESPONSE/AUTH_<TYPE>_LIST/AUTH_<TYPE>/IP_SET/IP_RANGE (#PCDATA)	
	A range of IP addresses saved in the authentication record.

XPath	element specifications / notes
/AUTH_<TYPE>_LIST_OUTPUT/RESPONSE/AUTH_<TYPE>_LIST/AUTH_<TYPE>/NETWORK_ID (#PCDATA)	
	The network ID for the record. Applies when the networks feature is enabled.
/AUTH_<TYPE>_LIST_OUTPUT/RESPONSE/AUTH_<TYPE>_LIST/AUTH_<TYPE>/CREATED (DATETIME|BY)	
/AUTH_<TYPE>_LIST_OUTPUT/RESPONSE/AUTH_<TYPE>_LIST/AUTH_<TYPE>/CREATED/DATETIME (#PCDATA)	
	The date and time the authentication record was created.
/AUTH_<TYPE>_LIST_OUTPUT/RESPONSE/AUTH_<TYPE>_LIST/AUTH_<TYPE>/CREATED/BY (#PCDATA)	
	The user login ID of the user who created the authentication record.
/AUTH_<TYPE>_LIST_OUTPUT/RESPONSE/AUTH_<TYPE>_LIST/AUTH_<TYPE>/LAST_MODIFIED (DATETIME)	
/AUTH_<TYPE>_LIST_OUTPUT/RESPONSE/AUTH_<TYPE>_LIST/AUTH_<TYPE>/LAST_MODIFIED/DATETIME (#PCDATA)	
	The date and time the authentication record was last modified.
/AUTH_<TYPE>_LIST_OUTPUT/RESPONSE/AUTH_<TYPE>_LIST/AUTH_<TYPE>/COMMENTS (#PCDATA)	
	User-provided notes (comments) saved in the record.

Authentication Record List by Type Output: Unix/Cisco IOS Response

Elements (in bold) for Unix and Cisco IOS records are described below.

XPath	element specifications / notes
/AUTH_UNIX_LIST_OUTPUT/RESPONSE/AUTH_UNIX_LIST/AUTH_UNIX	
	(ID, TITLE, **USERNAME, CLEARTEXT_PASSWORD, ROOT_TOOL, RSA_PRIVATE_KEY?, DSA_PRIVATE_KEY?, PORT?**, IP_SET, NETWORK_ID?, CREATED, LAST_MODIFIED, COMMENTS?, **USE_AGENTLESS_TRACKING?**, **AGENTLESS_TRACKING_PATH?**)
/AUTH_UNIX_LIST_OUTPUT/RESPONSE/AUTH_UNIX_LIST/AUTH_UNIX/USERNAME (#PCDATA)	
	The user account to be used for authentication on target hosts.
/AUTH_UNIX_LIST_OUTPUT/RESPONSE/AUTH_UNIX_LIST/AUTH_UNIX/CLEARTEXT_PASSWORD (#PCDATA)	
	A flag indicating whether the Cleartext Password option is enabled in the authentication record. The value 1 indicates that the option is enabled. The value 0 indicates that the option is disabled.
/AUTH_UNIX_LIST_OUTPUT/RESPONSE/AUTH_UNIX_LIST/AUTH_UNIX/ROOT_TOOL (#PCDATA)	
	A flag indicating whether a root delegation tool is enabled in the Unix authentication record: None (no root delegation tool is enabled), Sudo or PowerBroker.
/AUTH_UNIX_LIST_OUTPUT/RESPONSE/AUTH_UNIX_LIST/AUTH_UNIX/RSA_PRIVATE_KEY EMPTY	
	An RSA private key for the Unix authentication record.
/AUTH_UNIX_LIST_OUTPUT/RESPONSE/AUTH_UNIX_LIST/AUTH_UNIX/DSA_PRIVATE_KEY EMPTY	
	A DSA private key for the Unix authentication record.

XPath	element specifications / notes
/AUTH_UNIX_LIST_OUTPUT/RESPONSE/AUTH_UNIX_LIST/AUTH_UNIX/PORT (#PCDATA)	
	A list of custom ports defined for compliance scanning (authentication and compliance assessment).
/AUTH_UNIX_LIST_OUTPUT/RESPONSE/AUTH_UNIX_LIST/AUTH_UNIX/USE_AGENTLESS_TRACKING (#PCDATA)	
	A flag indicating whether Agentless Tracking is enabled for this record. The value 1 indicates that the option is enabled. The value 0 indicates that the option is disabled.
/AUTH_UNIX_LIST_OUTPUT/RESPONSE/AUTH_UNIX_LIST/AUTH_UNIX/AGENTLESS_TRACKING_PATH (#PCDATA)	
	The pathname where the host ID file will be stored on each host. (Applies only when Agentless Tracking is enabled in the record.)

Authentication Record List by Type Output: Windows Response

Windows-specific elements (in bold) are described below.

XPath	element specifications / notes
/AUTH_WINDOWS_LIST_OUTPUT/RESPONSE/AUTH_WINDOWS_LIST/AUTH_WINDOWS	
	(ID, TITLE, **USERNAME**, **NTLM?**, **WINDOWS_DOMAIN?**, **WINDOWS_AD_DOMAIN?**, **WINDOWS_AD_TRUST?**, IP_SET?, NETWORK_ID?, CREATED, LAST_MODIFIED, COMMENTS?, **USE_AGENTLESS_TRACKING?**)
/AUTH_WINDOWS_LIST_OUTPUT/RESPONSE/AUTH_WINDOWS_LIST/AUTH_WINDOWS /USERNAME (#PCDATA)	
	The user account to be used for authentication on target hosts.
/AUTH_WINDOWS_LIST_OUTPUT/RESPONSE/AUTH_WINDOWS_LIST/AUTH_WINDOWS/ NTLM (#PCDATA)	
	A flag indicating whether the NTLM authentication option is enabled in the authentication record. The value 1 indicates that the option is enabled. The value 0 indicates that the option is disabled.
/AUTH_WINDOWS_LIST_OUTPUT/RESPONSE/AUTH_WINDOWS_LIST/AUTH_WINDOWS/ WINDOWS_DOMAIN (#PCDATA)	
	A Windows domain name appears when a NetBIOS domain type is selected.
/AUTH_WINDOWS_LIST_OUTPUT/RESPONSE/AUTH_WINDOWS_LIST/AUTH_WINDOWS/ WINDOWS_AD_DOMAIN (#PCDATA)	
	An Active Directory domain name, specified as an FQDN name, appears when the Active Directory domain type is selected.
/AUTH_WINDOWS_LIST_OUTPUT/RESPONSE/AUTH_WINDOWS_LIST/AUTH_WINDOWS/ WINDOWS_AD_TRUST (#PCDATA)	
	A flag indicating whether the “Follow trust relationships” option is selected for an Active Directory domain. The value 1 indicates the “Follow trust relationships” option is enabled. The value 0 indicates the “Follow trust relationships” option is not enabled.

XPath	element specifications / notes
/AUTH_WINDOWS_LIST_OUTPUT/RESPONSE/AUTH_WINDOWS_LIST/AUTH_WINDOWS/USE_AGENTLESS_TRACKING (#PCDATA)	
	A flag indicating whether Agentless Tracking is enabled for this record. The value 1 indicates that the option is enabled. The value 0 indicates that the option is disabled.

Authentication Record List by Type Output: Oracle Response

Oracle-specific elements (in bold) are described below.

XPath	element specifications / notes
/AUTH_ORACLE_LIST_OUTPUT/RESPONSE/AUTH_ORACLE_LIST/AUTH_ORACLE	
	(ID, TITLE, **USERNAME, (SID|SERVICENAME), PORT, IP_SET, PC_ONLY?, WINDOWS_OS_CHECKS, WINDOWS_OS_OPTIONS?, UNIX_OPATCH_CHECKS, UNIX_OS_CHECKS, UNIX_OS_OPTIONS?**, NETWORK_ID?, CREATED, LAST_MODIFIED, COMMENTS?)
/AUTH_ORACLE_LIST_OUTPUT/RESPONSE/AUTH_ORACLE_LIST/AUTH_ORACLE/USERNAME (#PCDATA)	
	The user account to be used for authentication on target hosts.
/AUTH_ORACLE_LIST_OUTPUT/RESPONSE/AUTH_ORACLE_LIST/AUTH_ORACLE/SID (#PCDATA)	
	The Oracle System ID (SID) for the database instance to be authenticated to. This element appears only when a SID is defined for the Oracle record.
/AUTH_ORACLE_LIST_OUTPUT/RESPONSE/AUTH_ORACLE_LIST/AUTH_ORACLE/SERVICENAME (#PCDATA)	
	The Oracle service name for the database instance to be authenticated to. This element appears only when a service name is defined for the Oracle record.
/AUTH_ORACLE_LIST_OUTPUT/RESPONSE/AUTH_ORACLE_LIST/AUTH_ORACLE/PORT (#PCDATA)	
	The port number that the database instance is running on, if specified.
/AUTH_ORACLE_LIST_OUTPUT/RESPONSE/AUTH_ORACLE_LIST/AUTH_ORACLE/PC_ONLY (#PCDATA)	
	The value 1 indicates the **pc_only=1** parameter is specified for this record and this record is used for compliance scans only.
/AUTH_ORACLE_LIST_OUTPUT/RESPONSE/AUTH_ORACLE_LIST/AUTH_ORACLE/WINDOWS_OS_CHECKS (#PCDATA)	
	The value 1 indicates the option to perform Windows OS-level compliance checks is enabled for the record.
/AUTH_ORACLE_LIST_OUTPUT/RESPONSE/AUTH_ORACLE_LIST/AUTH_ORACLE/WINDOWS_OS_OPTIONS	
	(WIN_ORA_HOME, WIN_ORA_HOME_PATH, WIN_INIT_ORA_PATH, WIN_SPFILE_ORA_PATH, WIN_LISTENER_ORA_PATH, WIN_SQLNET_ORA_PATH, WIN_TNSNAMES_ORA_PATH)
	Values for Windows parameters used to perform OS-level compliance checks.

XPath	element specifications / notes
/AUTH_ORACLE_LIST_OUTPUT/RESPONSE/AUTH_ORACLE_LIST/AUTH_ORACLE/UNIX_OPATCH_CHECKS (#PCDATA)	
	The value 1 indicates the option to perform Unix OPatch compliance checks is enabled for the record.
/AUTH_ORACLE_LIST_OUTPUT/RESPONSE/AUTH_ORACLE_LIST/AUTH_ORACLE/UNIX_OS_CHECKS (#PCDATA)	
	The value 1 indicates the option to perform Unix OS-level compliance checks is enabled for the record.
/AUTH_ORACLE_LIST_OUTPUT/RESPONSE/AUTH_ORACLE_LIST/AUTH_ORACLE/UNIX_OS_OPTIONS	
	(UNIX_ORA_HOME_PATH, UNIX_INIT_ORA_PATH, UNIX_SPFILE_ORA_PATH, UNIX_LISTENER_ORA_PATH, UNIX_SQLNET_ORA_PATH, UNIX_TNSNAMES_ORA_PATH, UNIX_INVPTRLOC_PATH)
	Values for Unix parameters used to perform OS-level compliance checks.

Authentication Record List by Type Output: SNMP Response

SNMP-specific elements (in bold) are described below.

XPath	element specifications / notes
/AUTH_SNMP_LIST_OUTPUT/RESPONSE/AUTH_SNMP_LIST/AUTH_SNMP	
	(ID, TITLE, **USERNAME?, AUTH_ALG?, PRIV_ALG?, SEC_ENG?, CONTEXT_ENG?, CONTEXT?, COMMUNITY_STRINGS?,VERSION,** IP_SET, NETWORK_ID?, CREATED, LAST_MODIFIED, COMMENTS?)
/AUTH_SNMP_LIST_OUTPUT/RESPONSE/AUTH_SNMP_LIST/AUTH_SNMP/USERNAME (#PCDATA)	
	(SNMPv3 only) The user account to be used for authentication to target hosts.
/AUTH_SNMP_LIST_OUTPUT/RESPONSE/AUTH_SNMP_LIST/AUTH_SNMP/AUTH_ALG (#PCDATA)	
	(SNMPv3 only) The authentication algorithm to be used: SHA1 or MD5.
/AUTH_SNMP_LIST_OUTPUT/RESPONSE/AUTH_SNMP_LIST/AUTH_SNMP/PRIV_ALG (#PCDATA)	
	(SNMPv3 only) The algorithm to be used for privacy: DES or AES.
/AUTH_SNMP_LIST_OUTPUT/RESPONSE/AUTH_SNMP_LIST/AUTH_SNMP/SEC_ENG (#PCDATA)	
	(SNMPv3 only) The security engine ID.
/AUTH_SNMP_LIST_OUTPUT/RESPONSE/AUTH_SNMP_LIST/AUTH_SNMP/CONTEXT_ENG (#PCDATA)	
	(SNMPv3 only) The context engine ID.
/AUTH_SNMP_LIST_OUTPUT/RESPONSE/AUTH_SNMP_LIST/AUTH_SNMP/CONTEXT (#PCDATA)	
	(SNMPv3 only) The context name.
/AUTH_SNMP_LIST_OUTPUT/RESPONSE/AUTH_SNMP_LIST/AUTH_SNMP/COMMUNITY_STRINGS (#PCDATA)	
	(SNMPv1 or SNMPv2c only) User-provided SNMP community strings to be used for authentication to target hosts.

XPath	element specifications / notes
/AUTH_SNMP_LIST_OUTPUT/RESPONSE/AUTH_SNMP_LIST/AUTH_SNMP/VERSION (#PCDATA)	
	The SNMP protocol version: v1 (for SNMPv1), v2 (for SNMPv2c) or v3 (for SNMPv3).

Authentication Record List by Type Output: MS SQL Response

MS SQL-specific elements (in bold) are described below.

XPath	element specifications / notes
/AUTH_MS_SQL_LIST_OUTPUT/RESPONSE/AUTH_MS_SQL_LIST/AUTH_MS_SQL	
	(ID, TITLE, **USERNAME**, **(INSTANCE | AUTO_DISCOVER_INSTANCES), (DATABASE | AUTO_DISCOVER_DATABASES), (PORT | AUTO_DISCOVER_PORTS), DB_LOCAL**, **WINDOWS_DOMAIN?**, IP_SET, NETWORK_ID?, CREATED, LAST_MODIFIED, COMMENTS?)
/AUTH_MS_SQL_LIST_OUTPUT/RESPONSE/AUTH_MS_SQL_LIST/AUTH_MS_SQL/USERNAME (#PCDATA)	
	The user account to be used for authentication to target hosts.
/AUTH_MS_SQL_LIST_OUTPUT/RESPONSE/AUTH_MS_SQL_LIST/AUTH_MS_SQL/ INSTANCE|AUTO_DISCOVER_INSTANCES (#PCDATA)	
	A database instance or AUTO_DISCOVER_INSTANCES=1 if instances are auto-discovered.
/AUTH_MS_SQL_LIST_OUTPUT/RESPONSE/AUTH_MS_SQL_LIST/AUTH_MS_SQL/ DATABASE|AUTO_DISCOVER_DATABASES (#PCDATA)	
	A database name or AUTO_DISCOVER_DATABASES=1 if database names are auto-discovered.
/AUTH_MS_SQL_LIST_OUTPUT/RESPONSE/AUTH_MS_SQL_LIST/AUTH_MS_SQL/ PORT |AUTO_DISCOVER_PORTS (#PCDATA)	
	Port numbers or AUTO_DISSCOVER_PORTS=1 if ports are auto-discovered.
/AUTH_MS_SQL_LIST_OUTPUT/RESPONSE/AUTH_MS_SQL_LIST/AUTH_MS_SQL/ DB_LOCAL (#PCDATA)	
	A flag indicating the authentication type. Set o 1 when login credentials are for a MS SQL Server database account. Set to 0 when login credentials are for a Microsoft Windows operating system account that is associated with a MS SQL Server database account.
/AUTH_MS_SQL_LIST_OUTPUT/RESPONSE/AUTH_MS_SQL_LIST/AUTH_MS_SQL/ WINDOWS_DOMAIN (#PCDATA)	
	The domain name where the login credentials are stored, when the login credentials are for a Microsoft Windows operating system account.

Authentication Record List by Type Output: IBM DB2 Response

IBM DB2-specific elements (in bold) are described below.

XPath	element specifications / notes
/AUTH_IBM_DB2_LIST_OUTPUT/RESPONSE/AUTH_IBM_DB2_LIST/AUTH_IBM_DB2	(ID, TITLE, **USERNAME, DATABASE, PORT,** IP_SET, **PC_ONLY?**, NETWORK_ID?, CREATED, LAST_MODIFIED, COMMENTS?)
/AUTH_IBM_DB2_LIST_OUTPUT/RESPONSE/AUTH_IBM_DB2_LIST/AUTH_IBM_DB2/USERNAME (#PCDATA)	The user account to be used for authentication on target hosts.
/AUTH_IBM_DB2_LIST_OUTPUT/RESPONSE/AUTH_IBM_DB2_LIST/AUTH_IBM_DB2/DATABASE (#PCDATA)	The database name of the database to be scanned.
/AUTH_IBM_DB2_LIST_OUTPUT/RESPONSE/AUTH_IBM_DB2_LIST/AUTH_IBM_DB2/PORT (#PCDATA)	The port number that the database is running on.
/AUTH_IBM_DB2_LIST_OUTPUT/RESPONSE/AUTH_IBM_DB2_LIST/AUTH_IBM_DB2/PC_ONLY (#PCDATA)	The value 1 indicates the record is defined for compliance scans only.

Authentication Record List by Type Output: VMware Response

VMware-specific elements (in bold) are described below.

XPath	element specifications / notes
/AUTH_VMWARE_LIST_OUTPUT/RESPONSE/AUTH_VMWARE_LIST/AUTH_VMWARE	(ID, TITLE, **USERNAME, PORT, SSL_VERIFY, HOSTS?, IP_SET**, NETWORK_ID?, CREATED, LAST_MODIFIED, COMMENTS?)
/AUTH_VMWARE_LIST_OUTPUT/RESPONSE/AUTH_VMWARE_LIST/AUTH_VMWARE/USERNAME (#PCDATA)	The user account to be used for authentication on target hosts. This is an ESXi account or a Windows domain account, in which case the format is domain\user.
/AUTH_VMWARE_LIST_OUTPUT/RESPONSE/AUTH_VMWARE_LIST/AUTH_VMWARE/PORT (#PCDATA)	The port where the ESXi web services are running.
/AUTH_VMWARE_LIST_OUTPUT/RESPONSE/AUTH_VMWARE_LIST/AUTH_VMWARE/SSL_VERIFY (#PCDATA)	A flag indicating the SSL validation setting: "all" means complete SSL validation is selected, "skip" means the "Skip Verify" option is selected (host SSL certificate is self-signed or uses an SSL certificate signed by a custom root CA), "none" means no SSL validation is selected.
/AUTH_VMWARE_LIST_OUTPUT/RESPONSE/AUTH_VMWARE_LIST/AUTH_VMWARE/HOSTS (#PCDATA)	The list of FQDNs for hosts that correspond to all ESXi host IP addresses on which a custom SSL certificate signed by a trusted root CA is installed.

Authentication Record List by Type Output: Apache

Apache-specific elements (in bold) are described below.

XPath	element specifications / notes
/AUTH_APACHE_LIST_OUTPUT/RESPONSE/AUTH_APACHE_LIST/AUTH_APACHE	(ID, TITLE, IP_SET, **UNIX_CONFIGURATION_FILE**, **UNIX_CONTROL_COMMAND**, NETWORK_ID?, CREATED, LAST_MODIFIED, COMMENTS?)
/AUTH_APACHE_LIST_OUTPUT/RESPONSE/AUTH_APACHE_LIST/AUTH_APACHE/UNIX_CONFIGURATION_FILE (#PCDATA)	The path to the Apache configuration file (valid for Apache Web Server record only).
/AUTH_APACHE_LIST_OUTPUT/RESPONSE/AUTH_APACHE_LIST/AUTH_APACHE/UNIX_CONTROL_COMMAND (#PCDATA)	The path to the Apache control command (valid for Apache Web Server record only).

Authentication Record List by Type Output: IBM WebSphere

IBM WebSphere-specific elements (in bold) are described below.

XPath	element specifications / notes
/AUTH_APACHE_LIST_OUTPUT/RESPONSE/AUTH_APACHE_LIST/AUTH_APACHE	(ID, TITLE, IP_SET, **UNIX_INSTLLATION_DIRECTORY**, NETWORK_ID?, CREATED, LAST_MODIFIED, COMMENTS?)
/AUTH_APACHE_LIST_OUTPUT/RESPONSE/AUTH_IBM_WEBSPHERE_LIST/AUTH_IBM_WEBSPHERE/UNIX_INSTLLATION_DIRECTORY (#PCDATA)	The directory where the WebSphere application is installed.

Authentication Record List by Type Output: Tomcat Server

Tomcat Server-specific elements (in bold) are described below.

XPath	element specifications / notes
/AUTH_APACHE_LIST_OUTPUT/RESPONSE/AUTH_TOMCAT_LIST/AUTH_TOMCAT	(ID, TITLE, IP_SET, **INSTALLATION_PATH**, **INSTANCE_PATH?**, **AUTO_DISCOVER_INSTANCES?**, NETWORK_ID?, CREATED, LAST_MODIFIED, COMMENTS?)
/AUTH_APACHELIST_OUTPUT/RESPONSE/AUTH_TOMCAT_LIST/AUTH_TOMCAT/INSTALLATION_PATH (#PCDATA)	The directory where the tomcat server is installed.
/AUTH_APACHE_LIST_OUTPUT/RESPONSE/AUTH_TOMCAT_LIST/AUTH_TOMCAT/INSTANCE_PATH (#PCDATA)	The directory where the tomcat server instance(s) are installed, if specified.

XPath	element specifications / notes
/AUTH_APACHE_LIST_OUTPUT/RESPONSE/AUTH_TOMCAT_LIST/AUTH_TOMCAT/AUTO_DISCOVER_INSTANCES (#PCDATA)	
	The value 1 indicates that the "Auto Discover Instances" option is enabled for the record. The value 0 indicates that the option is disabled.

Authentication Record List by Type Output: HTTP Response

HTTP-specific elements (in bold) are described below.

XPath	element specifications / notes
/AUTH_HTTP_LIST_OUTPUT/RESPONSE/AUTH_HTTP_LIST/AUTH_HTTP	
	(ID, TITLE, **USERNAME, SSL, (REALM|VHOST)**, IP_SET?, NETWORK_ID?, CREATED, LAST_MODIFIED, COMMENTS?)
/AUTH_HTTP_LIST_OUTPUT/RESPONSE/AUTH_HTTP_LIST/AUTH_HTTP/USERNAME (#PCDATA)	
	The user name used for authentication.
/AUTH_HTTP_LIST_OUTPUT/RESPONSE/AUTH_HTTP_LIST/AUTH_HTTP/ SSL (#PCDATA)	
	A flag indicating the SSL setting. 1 means we'll attempt authentication over SSL only; 0 means we'll attempt authentication without this restriction.
/AUTH_HTTP_LIST_OUTPUT/RESPONSE/AUTH_HTTP_LIST/AUTH_HTTP/REALM (#PCDATA)	
	The realm to authenticate against.
/AUTH_HTTP_LIST_OUTPUT/RESPONSE/AUTH_HTTP_LIST/AUTH_HTTP/VHOST (#PCDATA)	
	The virtual host to authenticate against.

Authentication Record List by Type Output: MySQL Response

MySQL-specific elements (in bold) are described below.

XPath	element specifications / notes
/AUTH_HTTP_LIST_OUTPUT/RESPONSE/AUTH_MYSQL_LIST/AUTH_MYSQL	
	(ID, TITLE, **USERNAME**, **DATABASE**, **PORT**, **HOSTS?**, **IP_SET?**, **SSL_VERIFY**, **WINDOWS_CONF_FILE**, **UNIX_CONF_FILE**, **CLIENT_CERT?**, **CLIENT_KEY?**, NETWORK_ID?, CREATED, LAST_MODIFIED, COMMENTS?)
/AUTH_HTTP_LIST_OUTPUT/RESPONSE/AUTH_MYSQL_LIST/AUTH_MYSQL/USERNAME (#PCDATA)	
	The user name used for authentication.
/AUTH_HTTP_LIST_OUTPUT/RESPONSE/AUTH_MYSQL_LIST/AUTH_MYSQL/DATABASE (#PCDATA)	
	The database that will be authenticated to.
/AUTH_HTTP_LIST_OUTPUT/RESPONSE/AUTH_MYSQL_LIST/AUTH_MYSQL/PORT (#PCDATA)	
	The port the database is running on.
/AUTH_HTTP_LIST_OUTPUT/RESPONSE/AUTH_MYSQL_LIST/AUTH_MYSQL/HOSTS (#PCDATA)	
	A list of FQDNs for the hosts that correspond to all host API addresses on which a custom SSL certificate signed by a trusted root CA is installed.

XPath	element specifications / notes
/AUTH_HTTP_LIST_OUTPUT/RESPONSE/AUTH_MYSQL_LIST/AUTH_MYSQL/IP_SET (IP|IP_RANGE)	
	The IP address(es) the server will log into using the record's credentials.
/AUTH_HTTP_LIST_OUTPUT/RESPONSE/AUTH_MYSQL_LIST/AUTH_MYSQL/SSL_VERIFY (#PCDATA)	
	A flag indicating whether complete SSL certificate validation is enabled. The value 1 (enabled) means we'll send a login request after verifying that a connection the MySQL server uses SSL, the server SSL certificate is valid and matches the scanned host. The value 0 (disabled) means we'll attempt authentication with MySQL Servers that do and do not use SSL; in the case of SSL the server SSL certificate verification will be skipped.
/AUTH_HTTP_LIST_OUTPUT/RESPONSE/AUTH_MYSQL_LIST/AUTH_MYSQL/ WINDOWS_CONF_FILE (#PCDATA)	
	The path to the Windows MySQL conf file.
/AUTH_HTTP_LIST_OUTPUT/RESPONSE/AUTH_MYSQL_LIST/AUTH_MYSQL/ UNIX_CONF_FILE (#PCDATA)	
	The path to the Unix MySQL conf file.
/AUTH_HTTP_LIST_OUTPUT/RESPONSE/AUTH_MYSQL_LIST/AUTH_MYSQL/CLIENT_CERT (#PCDATA)	
	PEM-encoded X.509 certificate.
/AUTH_HTTP_LIST_OUTPUT/RESPONSE/AUTH_MYSQL_LIST/AUTH_MYSQL/CLIENT_KEY (#PCDATA)	
	PEM-encoded RSA private key.

Authentication Record List by Type Output: Warning List

Note that <TYPE> is the authentication type.

XPath	element specifications / notes
/AUTH_<TYPE>_LIST_OUTPUT/RESPONSE/WARNING_LIST (WARNING+)	
/AUTH_<TYPE>_LIST_OUTPUT/RESPONSE/WARNING_LIST/WARNING (CODE?, TEXT, URL?, ID_SET?)	
/AUTH_<TYPE>_LIST_OUTPUT/RESPONSE/WARNING_LIST/WARNING/CODE (#PCDATA)	
	A warning code. A warning code appears when the API request identifies more than 1,000 records (controls).
/AUTH_<TYPE>_LIST_OUTPUT/RESPONSE/WARNING_LIST/WARNING/TEXT (#PCDATA)	
	A warning message. A warning message appears when the API request identifies more than 1,000 records.
/AUTH_<TYPE>_LIST_OUTPUT/RESPONSE/WARNING_LIST/WARNING/URL (#PCDATA)	
	The URL for making another API request for the next batch of authentication records.
/AUTH_<TYPE>_LIST_OUTPUT/RESPONSE/WARNING_LIST/WARNING/ID_SET (ID|ID_RANGE)	
/AUTH_<TYPE>_LIST_OUTPUT/RESPONSE/WARNING_LIST/WARNING/ID_SET/ID (#PCDATA)	
	An authentication record ID.
/AUTH_<TYPE>_LIST_OUTPUT/RESPONSE/WARNING_LIST/WARNING/ID_SET/ID_RANGE (#PCDATA)	
	A range of authentication record IDs.

Authentication Record List by Type Output: Glossary

Note that <TYPE> is the authentication type, such as: unix, windows, oracle, snmp, etc.

XPath	element specifications / notes
/AUTH_<TYPE>_LIST_OUTPUT/RESPONSE/GLOSSARY (USER_LIST?)	
/AUTH_<TYPE>_LIST_OUTPUT/RESPONSE/GLOSSARY/USER_LIST (USER+)	
	A list of users who created authentication records in the authentication record list by type output.
/AUTH_<TYPE>_LIST_OUTPUT/RESPONSE/GLOSSARY/USER_LIST /USER	
	(USER_LOGIN, FIRST_NAME, LAST_NAME)
/AUTH_<TYPE>_LIST_OUTPUT/RESPONSE/GLOSSARY/USER_LIST /USER (#PCDATA)	
	A user login ID.
/AUTH_<TYPE>_LIST_OUTPUT/RESPONSE/GLOSSARY/USER_LIST /FIRST_NAME (#PCDATA)	
	The first name of the account user.
/AUTH_<TYPE>_LIST_OUTPUT/RESPONSE/GLOSSARY/USER_LIST /LAST_NAME (#PCDATA)	
	The last name of the account user.

Authentication Record Return

The generic return output XML is returned from an API request for an authentication record management request (create, edit or delete).

The DTD can be found at the following URL (where `qualysapi.qualys.com` is the API server URL where your account is located):

`https://qualysapi.qualys.com/api/2.0/batch_return.dtd`

DTD for Authentication Record Return Output

A recent DTD for the authentication record return output is shown below.

```
<!-- QUALYS BATCH_RETURN DTD -->
<!ELEMENT BATCH_RETURN (REQUEST?,RESPONSE)>

<!ELEMENT REQUEST (DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?,
                   POST_DATA?)>
<!ELEMENT DATETIME (#PCDATA)>
<!ELEMENT USER_LOGIN (#PCDATA)>
<!ELEMENT RESOURCE (#PCDATA)>
<!ELEMENT PARAM_LIST (PARAM+)>
<!ELEMENT PARAM (KEY, VALUE)>
<!ELEMENT KEY (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>
<!-- If specified, POST_DATA will be urlencoded -->
<!ELEMENT POST_DATA (#PCDATA)>

<!ELEMENT RESPONSE (DATETIME, BATCH_LIST?)>
<!ELEMENT BATCH_LIST (BATCH+)>
<!ELEMENT BATCH (CODE?, TEXT?, ID_SET?)>

<!ELEMENT CODE (#PCDATA)>
<!ELEMENT TEXT (#PCDATA)>
<!ELEMENT ID_SET (ID|ID_RANGE)+>
<!ELEMENT ID_RANGE (#PCDATA)>
<!ELEMENT ID (#PCDATA)>
<!-- EOF -->
```

XPaths for Authentication Record Return Output

This section describes the XPaths for the authentication record return output.

Authentication Record Return Output: Request

XPath	element specifications / notes
/BATCH_RETURN	(REQUEST?, RESPONSE)
/BATCH_RETURN/REQUEST	(DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?, POST_DATA?)
/BATCH_RETURN/REQUEST/DATETIME	(#PCDATA)
	The date and time of the API request.
/BATCH_RETURN/REQUEST/USER_LOGIN	(#PCDATA)
	The user login ID of the user who made the request.
/BATCH_RETURN/REQUEST/RESOURCE	(#PCDATA)
	The resource specified for the request.
/BATCH_RETURN/REQUEST/PARAM_LIST	(PARAM+)
/BATCH_RETURN/REQUEST/PARAM_LIST/PARAM	(KEY, VALUE)
/BATCH_RETURN/REQUEST/PARAM_LIST/PARAM/KEY	(#PCDATA)
	An input parameter name.
/BATCH_RETURN/REQUEST/PARAM_LIST/PARAM/VALUE	(#PCDATA)
	An input parameter value.
/BATCH_RETURN/REQUEST/POST_DATA	(#PCDATA)
	The POST data, if any. POST data is urlencoded.

Authentication Record Return Output: Response

XPath	element specifications / notes
/BATCH_RETURN	(REQUEST?, RESPONSE)
/BATCH_RETURN/RESPONSE	(DATETIME, BATCH_LIST?)
/BATCH_RETURN/RESPONSE/DATETIME	(#PCDATA)
	The date and time of the response.
/BATCH_RETURN/RESPONSE/BATCH_LIST	(BATCH+)
/BATCH_RETURN/RESPONSE/BATCH_LIST/BATCH	(CODE?, TEXT?, ID_SET?)
/BATCH_RETURN/RESPONSE/BATCH_LIST/BATCH/CODE	(#PCDATA)
	An HTTP status code indicating whether the operation succeeded or failed.
/BATCH_RETURN/RESPONSE/BATCH_LIST/BATCH/TEXT	(#PCDATA)
	An HTTP status message.
/BATCH_RETURN/RESPONSE/BATCH_LIST/BATCH/ID_SET	(ID|ID_RANGE)
	Record IDs or a range of IDs.

Authentication Vault List Output

The authentication vault list XML is returned from an authentication vault list API call (/api/2.0/fo/vault/ with action=list) API.

The DTD can be found at the following URL (where <qualysapi.qualys.com> is the API server URL where your account is located):

```
https://<qualysapi.qualys.com>/api/2.0/fo/vault/vault_output.dtd
```

DTD for Authentication Vault List Output

A recent DTD for the authentication vault list is shown below.

```
<!-- QUALYS VAULT_OUTPUT DTD -->

<!ELEMENT AUTH_VAULT_LIST_OUTPUT (REQUEST?,RESPONSE)>

<!ELEMENT REQUEST (DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?,
                   POST_DATA?)>
<!ELEMENT DATETIME (#PCDATA)>
<!ELEMENT USER_LOGIN (#PCDATA)>
<!ELEMENT RESOURCE (#PCDATA)>
<!ELEMENT PARAM_LIST (PARAM+)>
<!ELEMENT PARAM (KEY, VALUE)>
<!ELEMENT KEY (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>
<!-- if returned, POST_DATA will be urlencoded -->
<!ELEMENT POST_DATA (#PCDATA)>

<!ELEMENT RESPONSE (DATETIME, STATUS, COUNT, AUTH_VAULTS)>
<!ELEMENT STATUS (#PCDATA)>
<!ELEMENT COUNT (#PCDATA)>

<!ELEMENT AUTH_VAULTS (AUTH_VAULT*)>
<!ELEMENT AUTH_VAULT (UUID?, TITLE, VAULT_TYPE, LAST_MODIFIED?,
                      LAST_MODIFIED_DATE?, SERVER_ADDRESS?, ID?)>
<!ELEMENT UUID (#PCDATA)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT TITLE (#PCDATA)>
<!ELEMENT VAULT_TYPE (#PCDATA)>
<!ELEMENT SERVER_ADDRESS (#PCDATA)>
<!ELEMENT LAST_MODIFIED_DATE (#PCDATA)>
<!ELEMENT LAST_MODIFIED (DATETIME, BY)>
<!ELEMENT BY (#PCDATA)>
<!-- EOF -->
```

XPaths for Authentication Vault List Output

This section describes the XPaths for the authentication vault list output.

XPath	element specifications / notes
/AUTH_VAULT_LIST_OUTPUT	(REQUEST?, RESPONSE)
/AUTH_VAULT_LIST_OUTPUT/REQUEST	(DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?, POST_DATA?)
/AUTH_VAULT_LIST_OUTPUT/REQUEST/DATETIME	(#PCDATA) The date and time of the API request.
/AUTH_VAULT_LIST_OUTPUT/REQUEST/USER_LOGIN	(#PCDATA) The user login ID of the user who made the request.
/AUTH_VAULT_LIST_OUTPUT/REQUEST/RESOURCE	(#PCDATA) The resource specified for the request.
/AUTH_VAULT_LIST_OUTPUT/REQUEST/PARAM_LIST	(PARAM+)
/AUTH_VAULT_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM	(KEY, VALUE)
/AUTH_VAULT_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM/KEY	(#PCDATA) An input parameter name.
/AUTH_VAULT_LIST_OUTPUT/REQUEST/PARAM_LIST/PARAM/VALUE	(#PCDATA) An input parameter value.
/AUTH_VAULT_LIST_OUTPUT/REQUEST/POST_DATA	(#PCDATA) The POST data, if any.
/AUTH_VAULT_LIST_OUTPUT	(REQUEST?, RESPONSE)
/AUTH_VAULT_LIST_OUTPUT/RESPONSE	(DATETIME, STATUS, COUNT, AUTH_VAULTS)
/AUTH_VAULT_LIST_OUTPUT/RESPONSE/DATETIME	(#PCDATA) The date and time of the response.
/AUTH_VAULT_LIST_OUTPUT/RESPONSE/STATUS	(#PCDATA)
/AUTH_VAULT_LIST_OUTPUT/RESPONSE/COUNT	(#PCDATA)
/AUTH_VAULT_LIST_OUTPUT/RESPONSE/AUTH_VAULTS	(AUTH_VAULT*) (UUID?, TITLE, VAULT_TYPE, LAST_MODIFIED?, LAST_MODIFIED_DATE?, SERVER_ADDRESS?, ID?)
/AUTH_VAULT_LIST_OUTPUT/RESPONSE/AUTH_VAULTS/AUTH_VAULT/UUID	(#PCDATA) The UUID of the vault if available.
/AUTH_VAULT_LIST_OUTPUT/RESPONSE/AUTH_VAULTS/AUTH_VAULT/TITLE	(#PCDATA) The vault title.

XPath	element specifications / notes
/AUTH_VAULT_LIST_OUTPUT/RESPONSE/AUTH_VAULTS/AUTH_VAULT/VAULT_TYPE (#PCDATA)	
	The vault type, one of: Cyber-Ark PIM Suite, Thycotic Secret Server, Quest Vault, CA Access Control, Hitachi ID PAM, Lieberman ERPM
/AUTH_VAULT_LIST_OUTPUT/RESPONSE/AUTH_VAULTS/AUTH_VAULT/ LAST_MODIFIED (DATETIME, BY?)	
	The date/time the vault was last modified, and the username of the user who made the change.
/AUTH_VAULT_LIST_OUTPUT/RESPONSE/AUTH_VAULTS/AUTH_VAULT/ SERVER_ADDRESS (#PCDATA)	
	The IP address of the vault server. Valid for vault types: Cyber-Ark PIM Suite, Quest Vault.
/AUTH_VAULT_LIST_OUTPUT/RESPONSE/AUTH_VAULTS/AUTH_VAULT/ID (#PCDATA)	
	The vault ID.

Authentication Vault View Output

The authentication vault list XML is returned from an authentication vault list API call (/api/2.0/fo/vault/ with action=view) API.

The DTD can be found at the following URL (where <qualysapi.qualys.com> is the API server URL where your account is located):

```
https://<qualysapi.qualys.com>/api/2.0/fo/vault/vault_view.dtd
```

DTD for Authentication Vault View Output

A recent DTD for the authentication vault view is shown below.

```
<!-- QUALYS VAULT_OUTPUT DTD -->

<!ELEMENT VAULT_OUTPUT (REQUEST?,RESPONSE)>

<!ELEMENT REQUEST (DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?,
                   POST_DATA?)>
<!ELEMENT DATETIME (#PCDATA)>
<!ELEMENT USER_LOGIN (#PCDATA)>
<!ELEMENT RESOURCE (#PCDATA)>
<!ELEMENT PARAM_LIST (PARAM+)>
<!ELEMENT PARAM (KEY, VALUE)>
<!ELEMENT KEY (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>
<!-- if returned, POST_DATA will be urlencoded -->
<!ELEMENT POST_DATA (#PCDATA)>

<!ELEMENT RESPONSE (DATETIME, VAULT_QUEST)>

<!ELEMENT VAULT_QUEST (TITLE, COMMENTS, VAULT_TYPE, CREATED_ON?, OWNER?,
                       LAST_MODIFIED?, USERNAME?, URL?, SSL_VERIFY?,
                       DOMAIN?, API_USERNAME?, WEB_USERNAME?,
                       SERVER_ADDRESS?, PORT?, SAFE?, (UUID|ID))>
<!ELEMENT UUID (#PCDATA)>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT TITLE (#PCDATA)>
<!ELEMENT COMMENTS (#PCDATA)>
<!ELEMENT VAULT_TYPE (#PCDATA)>
<!ELEMENT CREATED_ON (#PCDATA)>
<!ELEMENT OWNER (#PCDATA)>
<!ELEMENT USERNAME (#PCDATA)>
<!ELEMENT URL (#PCDATA)>
<!ELEMENT SSL_VERIFY (#PCDATA)>
<!ELEMENT DOMAIN (#PCDATA)>
```

```
<!ELEMENT API_USERNAME (#PCDATA)>
<!ELEMENT WEB_USERNAME (#PCDATA)>
<!ELEMENT SERVER_ADDRESS (#PCDATA)>
<!ELEMENT PORT (#PCDATA)>
<!ELEMENT SAFE (#PCDATA)>
<!ELEMENT LAST_MODIFIED (DATETIME, BY?)>
<!ELEMENT BY (#PCDATA)>
<!-- EOF -->
```

XPaths for Authentication Vault View Output

This section describes the XPaths for the authentication vault view output.

XPath	element specifications / notes
/AUTH_VAULT_OUTPUT	(REQUEST?, RESPONSE)
/AUTH_VAULT_OUTPUT/REQUEST	(DATETIME, USER_LOGIN, RESOURCE, PARAM_LIST?, POST_DATA?)
/AUTH_VAULT_OUTPUT/REQUEST/DATETIME (#PCDATA)	The date and time of the API request.
/AUTH_VAULT_OUTPUT/REQUEST/USER_LOGIN (#PCDATA)	The user login ID of the user who made the request.
/AUTH_VAULT_OUTPUT/REQUEST/RESOURCE (#PCDATA)	The resource specified for the request.
/AUTH_VAULT_OUTPUT/REQUEST/PARAM_LIST (PARAM+)	
/AUTH_VAULT_OUTPUT/REQUEST/PARAM_LIST/PARAM (KEY, VALUE)	
/AUTH_VAULT_OUTPUT/REQUEST/PARAM_LIST/PARAM/KEY (#PCDATA)	An input parameter name.
/AUTH_VAULT_OUTPUT/REQUEST/PARAM_LIST/PARAM/VALUE (#PCDATA)	An input parameter value.
/AUTH_VAULT_OUTPUT/REQUEST/POST_DATA (#PCDATA)	The POST data, if any.
/AUTH_VAULT_OUTPUT	(REQUEST?, RESPONSE)
/AUTH_VAULT_OUTPUT/RESPONSE	(DATETIME, VAULT_QUEST)
/AUTH_VAULT_OUTPUT/RESPONSE/DATETIME (#PCDATA)	The date and time of the response.
/AUTH_VAULT_OUTPUT/RESPONSE/VAULT_QUEST	(TITLE, COMMENTS, VAULT_TYPE, CREATED_ON?, OWNER?, LAST_MODIFIED?, USERNAME?, URL?, SSL_VERIFY?, DOMAIN?, API_USERNAME?, WEB_USERNAME?, SERVER_ADDRESS?, PORT?, SAFE?, (UUID|ID))

XPath	element specifications / notes
/AUTH_VAULT_OUTPUT/RESPONSE/VAULT_QUEST/TITLE (#PCDATA)	
	The vault title.
/AUTH_VAULT_OUTPUT/RESPONSE/VAULT_QUEST/COMMENTS (#PCDATA)	
	User-defined comments for the vault.
/AUTH_VAULT_OUTPUT/RESPONSE/VAULT_QUEST/VAULT_TYPE (#PCDATA)	
	The vault type, one of: Cyber-Ark PIM Suite, Thycotic Secret Server, Quest Vault, CA Access Control, Hitachi ID PAM, Lieberman ERPM
/AUTH_VAULT_OUTPUT/RESPONSE/VAULT_QUEST/CREATED_ON (#PCDATA)	
	The date/time when the vault was first created.
/AUTH_VAULT_OUTPUT/RESPONSE/VAULT_QUEST/OWNER (#PCDATA)	
	The vault owner.
/AUTH_VAULT_OUTPUT/RESPONSE/VAULT_QUEST/LAST_MODIFIED (DATETIME, BY?)	
	The date/time when the vault was last modified and the username of the user who made the change.
/AUTH_VAULT_OUTPUT/RESPONSE/VAULT_QUEST/URL (#PCDATA)	
	The URL of the vault web services. Valid for vault types: CA Access Control, Lieberman ERPM, Thycotic Secret Server.
/AUTH_VAULT_OUTPUT/RESPONSE/VAULT_QUEST/SSL_VERIFY (#PCDATA)	
	A flag indicating whether our service will verify the SSL certificate of the web services URL to make sure the certificate is valid and trusted. Valid for vault types: CA Access Control, Lieberman ERPM, Thycotic Secret Server.
/AUTH_VAULT_OUTPUT/RESPONSE/VAULT_QUEST/DOMAIN (#PCDATA)	
	The domain name if your vault server is part of a domain. Valid vault types: Lieberman ERPM, Thycotic Secret Server.
/AUTH_VAULT_OUTPUT/RESPONSE/VAULT_QUEST/API_USERNAME (#PCDATA)	
	The username to be used for authentication to the vault.
/AUTH_VAULT_OUTPUT/RESPONSE/VAULT_QUEST/WEB_USERNAME (#PCDATA)	
	The web username to be used to access Basic authentication of the CA Access Control web server. Not valid for other vault t ypes.
/AUTH_VAULT_OUTPUT/RESPONSE/VAULT_QUEST/SERVER_ADDRESS (#PCDATA)	
	The IP address of the vault server. Valid for vault types: Cyber-Ark PIM Suite, Quest Vault.
/AUTH_VAULT_OUTPUT/RESPONSE/VAULT_QUEST/PORT (#PCDATA)	
	The port the vault server is running on. Valid for vault types: Cyber-Ark PIM Suite, Quest Vault.
/AUTH_VAULT_OUTPUT/RESPONSE/VAULT_QUEST/SAFE (#PCDATA)	
	The name of the digital password safe for Cyber-Ark PIM Suit vault. Not valid for other vault types.
/AUTH_VAULT_OUTPUT/RESPONSE/VAULT_QUEST/(UUID|ID) (#PCDATA)	
	The vault ID and UUID if available.

APPENDIX E

Scorecard Report XML

This appendix describes the XML output returned from API V2 requests using the Scorecard Report API functions.

- Asset Group Vulnerability Report
- Ignored Vulnerabilities Report
- Most Prevalent Vulnerabilities Report
- Most Vulnerable Hosts Report
- Patch Report

Asset Group Vulnerability Report

The asset group vulnerability report XML is returned from a scorecard report (**/report/scorecard**) API call.

The DTD can be found at the following URL:

`https://<qualysapi.qualys.com>/asset_group_scorecard.dtd`

where `<qualysapi.qualys.com>` is the API server URL where your account is located.

DTD for Asset Group Vulnerability Report

A recent DTD for the asset group vulnerability report is shown below.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- QUALYS ASSET GROUP SCORECARD DTD -->

<!ELEMENT ASSET_GROUP_SCORECARD (ERROR | (HEADER, SUMMARY, RESULTS))>
<!ELEMENT ERROR (#PCDATA)>
<!ATTLIST ERROR number CDATA #IMPLIED>

<!-- GENERIC HEADER -->
<!ELEMENT HEADER (NAME, GENERATION_DATETIME, COMPANY_INFO, USER_INFO)>
<!ELEMENT NAME (#PCDATA)>
<!ELEMENT GENERATION_DATETIME (#PCDATA)>
<!ELEMENT SCORECARD_TYPE (#PCDATA)>

<!ELEMENT COMPANY_INFO (NAME, ADDRESS, CITY, STATE, COUNTRY, ZIP_CODE)>
<!ELEMENT ADDRESS (#PCDATA)>
<!ELEMENT CITY (#PCDATA)>
<!ELEMENT STATE (#PCDATA)>
<!ELEMENT COUNTRY (#PCDATA)>
<!ELEMENT ZIP_CODE (#PCDATA)>

<!ELEMENT USER_INFO (NAME, USERNAME, ROLE)>
<!ELEMENT USERNAME (#PCDATA)>
<!ELEMENT ROLE (#PCDATA)>

<!-- TARGETING, FILTERING, SORTING CRITERIA -->
<!ELEMENT SUMMARY (PARAM_LIST, DETAILS?)>
<!ELEMENT PARAM_LIST (PARAM+)>
<!ELEMENT PARAM (KEY, VALUE)>
<!ELEMENT KEY (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>
```

```
<!-- RESULTS -->
<!ELEMENT RESULTS (ASSET_GROUP_LIST)>
<!ELEMENT ASSET_GROUP_LIST (ASSET_GROUP+)>
<!ELEMENT ASSET_GROUP (TITLE, STATS)>
<!ELEMENT TITLE (#PCDATA)>
<!ELEMENT STATS (HOSTS, NUM_SEV_5?, NUM_SEV_4?, NUM_SEV_3?,
                 HOSTS_543?, HOSTS_PCT_543?)>
<!ELEMENT HOSTS (#PCDATA)>
<!ELEMENT NUM_SEV_5 (#PCDATA)>
<!ELEMENT NUM_SEV_4 (#PCDATA)>
<!ELEMENT NUM_SEV_3 (#PCDATA)>
<!ELEMENT HOSTS_543 (#PCDATA)>
<!ELEMENT HOSTS_PCT_543 (#PCDATA)>
```

XPaths for Asset Group Vulnerability Report

The XPaths for the asset group vulnerability report are described below.

XPath	element specifications / notes
/ASSET_GROUP_SCORECARD	(ERROR | (HEADER, SUMMARY, RESULTS))
/ASSET_GROUP_SCORECARD/ERROR	(#PCDATA)
	An error message.
attribute: **number**	An error code, when available
/ASSET_GROUP_SCORECARD/HEADER	(NAME, GENERATION_DATETIME, COMPANY_INFO, USER_INFO)
/ASSET_GROUP_SCORECARD/HEADER/NAME	(#PCDATA)
	The report header name is "Asset Group Vulnerability Report".
/ASSET_GROUP_SCORECARD/HEADER/GENERATION_DATETIME	(#PCDATA)
	The date and time when the report was generated.
/ASSET_GROUP_SCORECARD/SCORECARD_TYPE	(#PCDATA)
	The scorecard type.
/ASSET_GROUP_SCORECARD/HEADER/COMPANY_INFO	(NAME, ADDRESS, CITY, STATE, COUNTRY, ZIP_CODE)
	The user's company name and address, as defined in the user's account.
/ASSET_GROUP_SCORECARD/HEADER/USER_INFO	(NAME, USERNAME, ROLE)
/ASSET_GROUP_SCORECARD/HEADER/USER_INFO/NAME	(#PCDATA)
	The name of the user who generated the scorecard.
/ASSET_GROUP_SCORECARD/HEADER/USER_INFO/USERNAME	(#PCDATA)
	The user login ID of the user who generated the scorecard.
/ASSET_GROUP_SCORECARD/HEADER/USER_INFO/ROLE	(#PCDATA)
	The user role assigned to the user who generated the scorecard: Manager, Unit Manager, Scanner or Reader.

XPath	element specifications / notes
/ASSET_GROUP_SCORECARD/SUMMARY	(PARAM_LIST, DETAILS?)
/ASSET_GROUP_SCORECARD/SUMMARY/PARAM_LIST	(PARAM+)
/ASSET_GROUP_SCORECARD/SUMMARY/PARAM_LIST/PARAM	(KEY, VALUE)
/ASSET_GROUP_SCORECARD/SUMMARY/PARAM_LIST/PARAM/KEY	(#PCDATA)
	A scorecard parameter name in the report source settings.
/ASSET_GROUP_SCORECARD/SUMMARY/PARAM_LIST/PARAM/VALUE	(#PCDATA)
	A scorecard parameter value in the report source settings.
/ASSET_GROUP_SCORECARD/RESULTS	(ASSET_GROUP_LIST)
/ASSET_GROUP_SCORECARD/RESULTS/ASSET_GROUP_LIST	(ASSET_GROUP+)
/ASSET_GROUP_SCORECARD/RESULTS/ASSET_GROUP	(TITLE, STATS)
/ASSET_GROUP_SCORECARD/RESULTS/ASSET_GROUP/TITLE	(#PCDATA)
	An asset group title.
/ASSET_GROUP_SCORECARD/RESULTS/ASSET_GROUP/STATS	(HOSTS, NUM_SEV_5?, NUM_SEV_4?, NUM_SEV_3?, HOSTS_543?, HOSTS_PCT_543?)
/ASSET_GROUP_SCORECARD/RESULTS/ASSET_GROUP/STATS/HOSTS	(#PCDATA)
	The number of live hosts in the asset group.
/ASSET_GROUP_SCORECARD/RESULTS/ASSET_GROUP/STATS/NUM_SEV_5	(#PCDATA)
	The number of severity 5 vulnerabilities currently active for the asset group.
/ASSET_GROUP_SCORECARD/RESULTS/ASSET_GROUP/STATS/NUM_SEV_4	(#PCDATA)
	The number of severity 4 vulnerabilities currently active for the asset group..
/ASSET_GROUP_SCORECARD/RESULTS/ASSET_GROUP/STATS/NUM_SEV_3	(#PCDATA)
	The number of severity 3 vulnerabilities currently active for the asset group.
/ASSET_GROUP_SCORECARD/RESULTS/ASSET_GROUP/STATS/HOSTS_543	(#PCDATA)
	The number of hosts in the asset group with at least one vulnerability that has a severity level of 5, 4, or 3.
/ASSET_GROUP_SCORECARD/RESULTS/ASSET_GROUP/STATS/HOSTS_PCT_543	(#PCDATA)
	The percentage of hosts in the asset group with at least one vulnerability that has a severity level of 5, 4, or 3.

Ignored Vulnerabilities Report

The ignored vulnerabilities report XML is returned from a scorecard report (**/report/scorecard**) API call.

The DTD can be found at the following URL:

`https://<qualysapi.qualys.com>/ignored_vulns_scorecard.dtd`

where `<qualysapi.qualys.com>` is the API server URL where your account is located.

DTD for Ignored Vulnerabilities Report

A recent DTD for the ignored vulnerabilities report is shown below.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- QUALYS IGNORED VULNS SCORECARD DTD -->

<!ELEMENT IGNORED_VULNS_SCORECARD (ERROR | (HEADER, SUMMARY, RESULTS))>
<!ELEMENT ERROR (#PCDATA)>
<!ATTLIST ERROR number CDATA #IMPLIED>

<!-- GENERIC HEADER -->
<!ELEMENT HEADER (NAME, GENERATION_DATETIME, COMPANY_INFO, USER_INFO)>
<!ELEMENT NAME (#PCDATA)>
<!ELEMENT GENERATION_DATETIME (#PCDATA)>
<!ELEMENT SCORECARD_TYPE (#PCDATA)>

<!ELEMENT COMPANY_INFO (NAME, ADDRESS, CITY, STATE, COUNTRY, ZIP_CODE)>
<!ELEMENT ADDRESS (#PCDATA)>
<!ELEMENT CITY (#PCDATA)>
<!ELEMENT STATE (#PCDATA)>
<!ELEMENT COUNTRY (#PCDATA)>
<!ELEMENT ZIP_CODE (#PCDATA)>

<!ELEMENT USER_INFO (NAME, USERNAME, ROLE)>
<!ELEMENT USERNAME (#PCDATA)>
<!ELEMENT ROLE (#PCDATA)>

<!-- TARGETING, FILTERING, SORTING CRITERIA -->
<!ELEMENT SUMMARY (PARAM_LIST)>
<!ELEMENT PARAM_LIST (PARAM+)>
<!ELEMENT PARAM (KEY, VALUE)>
<!ELEMENT KEY (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>
```

```
<!-- RESULTS -->
<!ELEMENT RESULTS (ASSET_GROUP_LIST)>
<!ELEMENT ASSET_GROUP_LIST (ASSET_GROUP+)>
<!ELEMENT ASSET_GROUP (TITLE, DETECTION_LIST)>

<!ELEMENT DETECTION_LIST (DETECTION+)>
<!ELEMENT DETECTION (HOST, VULN, TICKET)>

<!ELEMENT HOST (IP, DNS?, NETBIOS?, OS?)>
<!ELEMENT IP (#PCDATA)>
<!ELEMENT DNS (#PCDATA)>
<!ELEMENT NETBIOS (#PCDATA)>
<!ELEMENT OS (#PCDATA)>

<!ELEMENT VULN (QID, TITLE, FIRST_FOUND_DATE?, SEVERITY, TYPE,
               CVSS_BASE?, CVSS_TEMPORAL?)>
<!ELEMENT QID (#PCDATA)>
<!ELEMENT TITLE (#PCDATA)>
<!ELEMENT FIRST_FOUND_DATE (#PCDATA)>
<!ELEMENT SEVERITY (#PCDATA)>
<!ELEMENT TYPE (#PCDATA)>
<!ELEMENT CVSS_BASE (#PCDATA)>
<!ELEMENT CVSS_TEMPORAL (#PCDATA)>

<!ELEMENT TICKET (NUMBER, STATE_DAYS, LAST_MODIFIED_DATE, COMMENTS?,
                  ASSIGNEE_NAME?, ASSIGNEE_EMAIL?)>
<!ELEMENT NUMBER (#PCDATA)>
<!ELEMENT STATE_DAYS (#PCDATA)>
<!ELEMENT LAST_MODIFIED_DATE (#PCDATA)>
<!ELEMENT COMMENTS (#PCDATA)>
<!ELEMENT ASSIGNEE_NAME (#PCDATA)>
<!ELEMENT ASSIGNEE_EMAIL (#PCDATA)>
```

XPaths for Ignored Vulnerabilities Report

The XPaths for the ignored vulnerabilities report are described below.

XPath	element specifications / notes
/IGNORED_VULNS_SCORECARD	(ERROR | (HEADER, SUMMARY, RESULTS))
/IGNORED_VULNS_SCORECARD/ERROR (#PCDATA)	An error message.
attribute: **number**	An error code, when available
/IGNORED_VULNS_SCORECARD/HEADER	(NAME, GENERATION_DATETIME, COMPANY_INFO, USER_INFO)

XPath	element specifications / notes
/IGNORED_VULNS_SCORECARD/HEADER/NAME (#PCDATA)	
	The report header name is "Ignored Vulnerabilities Report".
/IGNORED_VULNS_SCORECARD/HEADER/GENERATION_DATETIME (#PCDATA)	
	The date and time when the report was generated.
/IGNORED_VULNS_SCORECARD/HEADER/SCORECARD_TYPE (#PCDATA)	
	The scorecard type.
/IGNORED_VULNS_SCORECARD/HEADER/COMPANY_INFO	
	(NAME, ADDRESS, CITY, STATE, COUNTRY, ZIP_CODE)
	The user's company name and address, as defined in the user's account.
/IGNORED_VULNS_SCORECARD/HEADER/USER_INFO (NAME, USERNAME, ROLE)	
/IGNORED_VULNS_SCORECARD/HEADER/USER_INFO/NAME (#PCDATA)	
	The name of the user who generated the scorecard.
/IGNORED_VULNS_SCORECARD/HEADER/USER_INFO/USERNAME (#PCDATA)	
	The user login ID of the user who generated the scorecard.
/IGNORED_VULNS_SCORECARD/HEADER/USER_INFO/ROLE (#PCDATA)	
	The user role assigned to the user who generated the scorecard: Manager, Unit Manager, Scanner or Reader..
/IGNORED_VULNS_SCORECARD/SUMMARY (PARAM_LIST)	
/IGNORED_VULNS_SCORECARD/SUMMARY/PARAM_LIST (PARAM+)	
/IGNORED_VULNS_SCORECARD/SUMMARY/PARAM_LIST/PARAM (KEY, VALUE)	
/IGNORED_VULNS_SCORECARD/SUMMARY/PARAM_LIST/PARAM/KEY (#PCDATA)	
	A scorecard parameter name in the report source settings.
/IGNORED_VULNS_SCORECARD/SUMMARY/PARAM_LIST/PARAM/VALUE (#PCDATA)	
	A scorecard parameter value in the report source settings.
/IGNORED_VULNS_SCORECARD/RESULTS (ASSET_GROUP_LIST)	
/IGNORED_VULNS_SCORECARD/RESULTS/ASSET_GROUP_LIST (ASSET_GROUP+)	
/IGNORED_VULNS_SCORECARD/RESULTS/ASSET_GROUP_LIST/ASSET_GROUP (TITLE, DETECTION_LIST)	
/IGNORED_VULNS_SCORECARD/RESULTS/ASSET_GROUP_LIST/ASSET_GROUP/TITLE	
	An asset group title.
/IGNORED_VULNS_SCORECARD/RESULTS/ASSET_GROUP_LIST/ASSET_GROUP/DETECTION_LIST	
	(DETECTION+)
/IGNORED_VULNS_SCORECARD/RESULTS/ASSET_GROUP_LIST/ASSET_GROUP/DETECTION_LIST/DETECTION (HOST, VULN, TICKET)	
/IGNORED_VULNS_SCORECARD/RESULTS/ASSET_GROUP_LIST/ASSET_GROUP/DETECTION_LIST/DETECTION/HOST (IP, DNS?, NETBIOS?, OS?)	
	Information about the host, including its IP address and this additional information when available: DNS hostname, NetBIOS hostname, and operating system.

XPath	element specifications / notes
/IGNORED_VULNS_SCORECARD/RESULTS/ASSET_GROUP_LIST/ASSET_GROUP/DETECTION_LIST/DETECTION/VULN	
	(QID, TITLE, FIRST_FOUND_DATE?, SEVERITY, TYPE, CVSS_BASE?, CVSS_TEMPORAL?)
	Information about the vulnerability detected. CVSS Base and Temporal scores are included when the CVSS Scoring feature is enabled for the subscription.
/IGNORED_VULNS_SCORECARD/RESULTS/ASSET_GROUP_LIST/ASSET_GROUP/DETECTION_LIST/DETECTION/TICKET	
	(NUMBER, STATE_DAYS, LAST_MODIFIED_DATE, COMMENTS?, ASSIGNEE_NAME?, ASSIGNEE_EMAIL?)
	Information about a related ticket if one exists. Information includes the ticket number, the number of days the ticket has been in the Closed/Ignored state, and the date the ticket was created or last modified, any user-defined comments, and the ticket assignee's name and email address.

Most Prevalent Vulnerabilities Report

The most prevalent vulnerabilities report XML is returned from a scorecard report (**`/report/scorecard`**) API call.

The DTD can be found at the following URL:

`https://<qualysapi.qualys.com>/most_prevalent_vulns_scorecard.dtd`

where `<qualysapi.qualys.com>` is the API server URL where your account is located.

DTD for Most Prevalent Vulnerabilities Report

A recent DTD for the most prevalent vulnerabilities report is shown below.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- QUALYS MOST PREVALENT VULNS SCORECARD DTD -->

<!ELEMENT MOST_PREVALENT_VULNS_SCORECARD (ERROR | (HEADER, SUMMARY,
                                          RESULTS))>
<!ELEMENT ERROR (#PCDATA)>
<!ATTLIST ERROR number CDATA #IMPLIED>

<!-- GENERIC HEADER -->
<!ELEMENT HEADER (NAME, GENERATION_DATETIME, COMPANY_INFO, USER_INFO)>
<!ELEMENT NAME (#PCDATA)>
<!ELEMENT GENERATION_DATETIME (#PCDATA)>
<!ELEMENT SCORECARD_TYPE (#PCDATA)>

<!ELEMENT COMPANY_INFO (NAME, ADDRESS, CITY, STATE, COUNTRY, ZIP_CODE)>
<!ELEMENT ADDRESS (#PCDATA)>
<!ELEMENT CITY (#PCDATA)>
<!ELEMENT STATE (#PCDATA)>
<!ELEMENT COUNTRY (#PCDATA)>
<!ELEMENT ZIP_CODE (#PCDATA)>

<!ELEMENT USER_INFO (NAME, USERNAME, ROLE)>
<!ELEMENT USERNAME (#PCDATA)>
<!ELEMENT ROLE (#PCDATA)>

<!-- TARGETING, FILTERING, SORTING CRITERIA -->
<!ELEMENT SUMMARY (PARAM_LIST, DETAILS?)>
<!ELEMENT PARAM_LIST (PARAM+)>
<!ELEMENT PARAM (KEY, VALUE)>
<!ELEMENT KEY (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>
```

```
<!-- RESULTS -->
<!ELEMENT RESULTS (VULN_LIST)>
<!ELEMENT VULN_LIST (VULN+)>
<!ELEMENT VULN (RANK, QID, TITLE, SEVERITY, TYPE, FIRST_FOUND_DATE?,
                DETECTIONS?, CVSS_BASE?, CVSS_TEMPORAL?,
                TOTAL_HOSTS_AFFECTED?, PERCENT_HOSTS_AFFECTED?)>
<!ELEMENT RANK (#PCDATA)>
<!ELEMENT QID (#PCDATA)>
<!ELEMENT TITLE (#PCDATA)>
<!ELEMENT SEVERITY (#PCDATA)>
<!ELEMENT TYPE (#PCDATA)>
<!ELEMENT FIRST_FOUND_DATE (#PCDATA)>
<!ELEMENT DETECTIONS (#PCDATA)>
<!ELEMENT CVSS_BASE (#PCDATA)>
<!ELEMENT CVSS_TEMPORAL (#PCDATA)>
<!ELEMENT TOTAL_HOSTS_AFFECTED (#PCDATA)>
<!ELEMENT PERCENT_HOSTS_AFFECTED (#PCDATA)>
```

XPaths for Most Prevalent Vulnerabilities Report

The XPaths for the most prevalent vulnerabilities report are described below.

XPath	element specifications / notes
/MOST_PREVALENT_VULNS_SCORECARD	(ERROR | (HEADER, SUMMARY, RESULTS))
/MOST_PREVALENT_VULNS_SCORECARD/ERROR (#PCDATA)	An error message.
attribute: **number**	An error code, when available
/MOST_PREVALENT_VULNS_SCORECARD/HEADER	(NAME, GENERATION_DATETIME, COMPANY_INFO, USER_INFO)
/MOST_PREVALENT_VULNS_SCORECARD/HEADER/NAME (#PCDATA)	The report header name is “Most Prevalent Vulnerabilities Report”.
/MOST_PREVALENT_VULNS_SCORECARD/HEADER/GENERATION_DATETIME (#PCDATA)	The date and time when the report was generated.
/MOST_PREVALENT_VULNS_SCORECARD/HEADER/SCORECARD_TYPE (#PCDATA)	The scorecard type.
/MOST_PREVALENT_VULNS_SCORECARD/HEADER/COMPANY_INFO	(NAME, ADDRESS, CITY, STATE, COUNTRY, ZIP_CODE)
	The user’s company name and address, as defined in the user’s account.
/MOST_PREVALENT_VULNS_SCORECARD/HEADER/USER_INFO (NAME, USERNAME, ROLE)	
/MOST_PREVALENT_VULNS_SCORECARD/HEADER/USER_INFO/NAME (#PCDATA)	The name of the user who generated the scorecard.

XPath	element specifications / notes
/MOST_PREVALENT_VULNS_SCORECARD/HEADER/USER_INFO/USERNAME (#PCDATA)	
	The user login ID of the user who generated the scorecard.
/MOST_PREVALENT_VULNS_SCORECARD/HEADER/USER_INFO/ROLE (#PCDATA)	
	The user role assigned to the user who generated the scorecard: Manager, Unit Manager, Scanner or Reader.
/MOST_PREVALENT_VULNS_SCORECARD/SUMMARY (PARAM_LIST)	
/MOST_PREVALENT_VULNS_SCORECARD/SUMMARY/PARAM_LIST (PARAM+)	
/MOST_PREVALENT_VULNS_SCORECARD/SUMMARY/PARAM_LIST/PARAM (KEY, VALUE)	
/MOST_PREVALENT_VULNS_SCORECARD/SUMMARY/PARAM_LIST/PARAM/KEY (#PCDATA)	
	A scorecard parameter name in the report source settings.
/MOST_PREVALENT_VULNS_SCORECARD/SUMMARY/PARAM_LIST/PARAM/VALUE (#PCDATA)	
	A scorecard parameter value in the report source settings.
/MOST_PREVALENT_VULNS_SCORECARD/RESULTS (VULN_LIST)	
/MOST_PREVALENT_VULNS_SCORECARD/RESULTS/VULN_LIST (VULN+)	
/MOST_PREVALENT_VULNS_SCORECARD/RESULTS/VULN	
	(RANK, QID, TITLE, SEVERITY, TYPE, FIRST_FOUND_DATE?, DETECTIONS?, CVSS_BASE?, CVSS_TEMPORAL?, TOTAL_HOSTS_AFFECTED?, PERCENT_HOSTS_AFFECTED?)
/MOST_PREVALENT_VULNS_SCORECARD/RESULTS/VULN/RANK (#PCDATA)	
	The rank of the vulnerability. The vulnerability that was detected on the largest number of hosts is listed as #1.
/MOST_PREVALENT_VULNS_SCORECARD/RESULTS/VULN/QID (#PCDATA)	
	The QID assigned to the vulnerability.
/MOST_PREVALENT_VULNS_SCORECARD/RESULTS/VULN/TITLE (#PCDATA)	
	The vulnerability title.
/MOST_PREVALENT_VULNS_SCORECARD/RESULTS/VULN/SEVERITY (#PCDATA)	
	The severity level assigned to the vulnerability.
/MOST_PREVALENT_VULNS_SCORECARD/RESULTS/VULN/TYPE (#PCDATA)	
	The vulnerability type.
/MOST_PREVALENT_VULNS_SCORECARD/RESULTS/VULN/FIRST_FOUND_DATE (#PCDATA)	
	The date and time the vulnerability was first detected.
/MOST_PREVALENT_VULNS_SCORECARD/RESULTS/VULN/DETECTIONS (#PCDATA)	
	The total number of times the vulnerability was detected.
/MOST_PREVALENT_VULNS_SCORECARD/RESULTS/VULN/CVSS_BASE (#PCDATA)	
	The CVSS base score for the vulnerability. This is displayed only when the CVSS Scoring feature is enabled for the subscription.
/MOST_PREVALENT_VULNS_SCORECARD/RESULTS/VULN/CVSS_TEMPORAL (#PCDATA)	
	The CVSS temporal score for the vulnerability. This is displayed only when the CVSS Scoring feature is enabled for the subscription.

XPath	element specifications / notes
/MOST_PREVALENT_VULNS_SCORECARD/RESULTS/VULN/TOTAL_HOSTS_AFFECTED (#PCDATA)	
	The number of hosts that are currently affected by the vulnerability.
/MOST_PREVALENT_VULNS_SCORECARD/RESULTS/VULN/PERCENT_HOSTS_AFFECTED (#PCDATA)	
	The percentage of hosts that are currently affected by the vulnerability.

Most Vulnerable Hosts Report

The most vulnerable hosts report XML is returned from a scorecard report (**/report/scorecard**) API call.

The DTD can be found at the following URL:

`https://<qualysapi.qualys.com>/most_vulnerable_hosts_scorecard.dtd`

where `<qualysapi.qualys.com>` is the API server URL where your account is located.

DTD for Most Vulnerable Hosts Report

A recent DTD for the most vulnerable hosts report is shown below.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- QUALYS MOST VULNERABLE HOSTS SCORECARD DTD -->

<!ELEMENT MOST_VULNERABLE_HOSTS_SCORECARD (ERROR | (HEADER, SUMMARY,
                                                    RESULTS))>
<!ELEMENT ERROR (#PCDATA)>
<!ATTLIST ERROR number CDATA #IMPLIED>

<!-- GENERIC HEADER -->
<!ELEMENT HEADER (NAME, GENERATION_DATETIME, COMPANY_INFO, USER_INFO)>
<!ELEMENT NAME (#PCDATA)>
<!ELEMENT GENERATION_DATETIME (#PCDATA)>
<!ELEMENT SCORECARD_TYPE (#PCDATA)>

<!ELEMENT COMPANY_INFO (NAME, ADDRESS, CITY, STATE, COUNTRY, ZIP_CODE)>
<!ELEMENT ADDRESS (#PCDATA)>
<!ELEMENT CITY (#PCDATA)>
<!ELEMENT STATE (#PCDATA)>
<!ELEMENT COUNTRY (#PCDATA)>
<!ELEMENT ZIP_CODE (#PCDATA)>

<!ELEMENT USER_INFO (NAME, USERNAME, ROLE)>
<!ELEMENT USERNAME (#PCDATA)>
<!ELEMENT ROLE (#PCDATA)>

<!-- TARGETING, FILTERING, SORTING CRITERIA -->
<!ELEMENT SUMMARY (PARAM_LIST, DETAILS?)>
<!ELEMENT PARAM_LIST (PARAM+)>
<!ELEMENT PARAM (KEY, VALUE)>
<!ELEMENT KEY (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>
```

```
<!-- RESULTS -->
<!ELEMENT RESULTS (HOST_LIST)>
<!ELEMENT HOST_LIST (HOST+)>
<!ELEMENT HOST (RANK, IP, DNS?, NETBIOS?, LAST_SCAN_DATE?,
                NUM_SEV_5, NUM_SEV_4, BUSINESS_RISK, SECURITY_RISK,
                ASSET_GROUPS?)>
<!ELEMENT RANK (#PCDATA)>
<!ELEMENT IP (#PCDATA)>
<!ELEMENT DNS (#PCDATA)>
<!ELEMENT NETBIOS (#PCDATA)>
<!ELEMENT LAST_SCAN_DATE (#PCDATA)>
<!ELEMENT NUM_SEV_5 (#PCDATA)>
<!ELEMENT NUM_SEV_4 (#PCDATA)>
<!ELEMENT BUSINESS_RISK (#PCDATA)>
<!ELEMENT SECURITY_RISK (#PCDATA)>
<!ELEMENT ASSET_GROUPS (#PCDATA)>
```

XPaths for Most Vulnerable Hosts Report

The XPaths for the most vulnerable hosts report are described below.

XPath	element specifications / notes
/MOST_VULNERABLE_HOSTS_SCORECARD	(ERROR | (HEADER, SUMMARY, RESULTS))
/MOST_VULNERABLE_HOSTS_SCORECARD/ERROR (#PCDATA)	An error message.
attribute: **number**	An error code, when available
/MOST_VULNERABLE_HOSTS_SCORECARD/HEADER	(NAME, GENERATION_DATETIME, COMPANY_INFO, USER_INFO)
/MOST_VULNERABLE_HOSTS_SCORECARD/HEADER/NAME (#PCDATA)	The report header name is “Most Vulnerable Hosts Report”.
/MOST_VULNERABLE_HOSTS_SCORECARD/HEADER/GENERATION_DATETIME (#PCDATA)	The date and time when the report was generated.
/MOST_VULNERABLE_HOSTS_SCORECARD/HEADER/SCORECARD_TYPE (#PCDATA)	The scorecard type.
/MOST_VULNERABLE_HOSTS_SCORECARD/HEADER/COMPANY_INFO	(NAME, ADDRESS, CITY, STATE, COUNTRY, ZIP_CODE)
	The user’s company name and address, as defined in the user’s account.
/MOST_VULNERABLE_HOSTS_SCORECARD/HEADER/USER_INFO (NAME, USERNAME, ROLE)	
/MOST_VULNERABLE_HOSTS_SCORECARD/HEADER/USER_INFO/NAME (#PCDATA)	The name of the user who generated the scorecard.

XPath	element specifications / notes
/MOST_VULNERABLE_HOSTS_SCORECARD/HEADER/USER_INFO/USERNAME (#PCDATA)	
	The user login ID of the user who generated the scorecard.
/MOST_VULNERABLE_HOSTS_SCORECARD/HEADER/USER_INFO/ROLE (#PCDATA)	
	The user role assigned to the user who generated the scorecard: Manager, Unit Manager, Scanner or Reader.
/MOST_VULNERABLE_HOSTS_SCORECARD/SUMMARY (PARAM_LIST)	
/MOST_VULNERABLE_HOSTS_SCORECARD/SUMMARY/PARAM_LIST (PARAM+)	
/MOST_VULNERABLE_HOSTS_SCORECARD/SUMMARY/PARAM_LIST/PARAM (KEY, VALUE)	
/MOST_VULNERABLE_HOSTS_SCORECARD/SUMMARY/PARAM_LIST/PARAM/KEY (#PCDATA)	
	A scorecard parameter name in the report source settings.
/MOST_VULNERABLE_HOSTS_SCORECARD/SUMMARY/PARAM_LIST/PARAM/VALUE (#PCDATA)	
	A scorecard parameter value in the report source settings.
/MOST_VULNERABLE_HOSTS_SCORECARD/RESULTS (HOST_LIST)	
/MOST_VULNERABLE_HOSTS_SCORECARD/HOST_LIST (HOST+)	
/MOST_VULNERABLE_HOSTS_SCORECARD/HOST	
	(RANK, IP, DNS?, NETBIOS?, LAST_SCAN_DATE?, NUM_SEV_5, NUM_SEV_4, BUSINESS_RISK, SECURITY_RISK, ASSET_GROUPS?)
/MOST_VULNERABLE_HOSTS_SCORECARD/HOST/RANK (#PCDATA)	
	The rank for the host. The host with the highest number of vulnerabilities with severity levels 4 and 5 is listed as #1.
/MOST_VULNERABLE_HOSTS_SCORECARD/HOST/IP (#PCDATA)	
	The IP address for the host.
/MOST_VULNERABLE_HOSTS_SCORECARD/HOST/DNS (#PCDATA)	
	The DNS hostname.
/MOST_VULNERABLE_HOSTS_SCORECARD/HOST/NETBIOS (#PCDATA)	
	The NetBIOS hostname.
/MOST_VULNERABLE_HOSTS_SCORECARD/HOST/LAST_SCAN_DATE (#PCDATA)	
	The date and time the host was last scanned for vulnerabilities.
/MOST_VULNERABLE_HOSTS_SCORECARD/HOST/NUM_SEV_5 (#PCDATA)	
	The current number of severity 5 vulnerabilities detected on the host.
/MOST_VULNERABLE_HOSTS_SCORECARD/HOST/NUM_SEV_4 (#PCDATA)	
	The current number of severity 4 vulnerabilities detected on the host.
/MOST_VULNERABLE_HOSTS_SCORECARD/HOST/BUSINESS_RISK (#PCDATA)	
	The business risk value. See "Business Risk" in the online help for information. If the host belongs to one asset group in the report, the business risk value for that asset group is displayed. If the host belongs to multiple asset groups in the report, the highest business risk value across the asset groups is displayed.

XPath	element specifications / notes
/MOST_VULNERABLE_HOSTS_SCORECARD/HOST/SECURITY_RISK (#PCDATA)	
	The highest severity level across the vulnerabilities and potential vulnerabilities detected on the host.
/MOST_VULNERABLE_HOSTS_SCORECARD/HOST/ASSET_GROUPS (#PCDATA)	
	A list of asset groups that the host belongs to.

Patch Report

The patch report XML is returned from a scorecard report (**/report/scorecard**) API call.

The DTD can be found at the following URL:

`https://<qualysapi.qualys.com>/patch_scorecard.dtd`

where `<qualysapi.qualys.com>` is the API server URL where your account is located.

DTD for Patch Report

A recent DTD for the patch report is shown below.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- QUALYS PATCH REPORT SCORECARD DTD -->

<!ELEMENT PATCH_REPORT_SCORECARD (ERROR | (HEADER, SUMMARY, RESULTS))>
<!ELEMENT ERROR (#PCDATA)>
<!ATTLIST ERROR number CDATA #IMPLIED>

<!-- GENERIC HEADER -->
<!ELEMENT HEADER (NAME, GENERATION_DATETIME, COMPANY_INFO, USER_INFO)>
<!ELEMENT NAME (#PCDATA)>
<!ELEMENT GENERATION_DATETIME (#PCDATA)>

<!ELEMENT COMPANY_INFO (NAME, ADDRESS, CITY, STATE, COUNTRY, ZIP_CODE)>
<!ELEMENT ADDRESS (#PCDATA)>
<!ELEMENT CITY (#PCDATA)>
<!ELEMENT STATE (#PCDATA)>
<!ELEMENT COUNTRY (#PCDATA)>
<!ELEMENT ZIP_CODE (#PCDATA)>

<!ELEMENT USER_INFO (NAME, USERNAME, ROLE)>
<!ELEMENT USERNAME (#PCDATA)>
<!ELEMENT ROLE (#PCDATA)>

<!-- TARGETING, FILTERING, SORTING CRITERIA -->
<!ELEMENT SUMMARY (PARAM_LIST, DETAILS?)>
<!ELEMENT PARAM_LIST (PARAM+)>
<!ELEMENT PARAM (KEY, VALUE)>
<!ELEMENT KEY (#PCDATA)>
<!ELEMENT VALUE (#PCDATA)>

<!-- SUMMARY DETAILS -->
<!ELEMENT DETAILS (ASSET_GROUP_LIST)>
```

```
<!ELEMENT ASSET_GROUP_LIST (ASSET_GROUP*)>
<!ELEMENT ASSET_GROUP (TITLE, (STATS | DETECTION_LIST))>
<!ELEMENT STATS (NUM_HOSTS?, SCANNED_HOSTS?, MISSING?)>
<!ELEMENT NUM_HOSTS (#PCDATA)>
<!ELEMENT SCANNED_HOSTS (#PCDATA)>
<!ELEMENT MISSING (ONE_OR_MORE_PATCHES?, SOFTWARE_1?, SOFTWARE_2?)>
<!ELEMENT ONE_OR_MORE_PATCHES (PERCENT, TOTAL_HOSTS)>
<!ELEMENT SOFTWARE_1 (PERCENT, TOTAL_HOSTS, QID?)>
<!ELEMENT SOFTWARE_2 (PERCENT, TOTAL_HOSTS, QID?)>
<!ELEMENT PERCENT (#PCDATA)>
<!ELEMENT TOTAL_HOSTS (#PCDATA)>
<!ELEMENT QID (#PCDATA)>

<!-- RESULTS -->
<!ELEMENT RESULTS (ASSET_GROUP_LIST)>

<!ELEMENT DETECTION_LIST (DETECTION*)>
<!ELEMENT DETECTION (HOST, VULN)>

<!ELEMENT HOST (IP, DNS?, NETBIOS?, OS?, OWNER?)>
<!ELEMENT IP (#PCDATA)>
<!ELEMENT DNS (#PCDATA)>
<!ELEMENT NETBIOS (#PCDATA)>
<!ELEMENT OS (#PCDATA)>
<!ELEMENT OWNER (#PCDATA)>

<!ELEMENT VULN (QID, VENDOR_REF?, TITLE)>
<!ELEMENT VENDOR_REF (#PCDATA)>
<!ELEMENT TITLE (#PCDATA)>
```

XPaths for Patch Report

The XPaths for the patch report are described below.

XPath	element specifications / notes
/PATCH_REPORT_SCORECARD	(ERROR | (HEADER, SUMMARY, RESULTS))
/PATCH_REPORT_SCORECARD/ERROR (#PCDATA)	An error message.
attribute: **number**	An error code, when available
/PATCH_REPORT_SCORECARD/HEADER	(NAME, GENERATION_DATETIME, COMPANY_INFO, USER_INFO)
/PATCH_REPORT_SCORECARD/HEADER/NAME (#PCDATA)	The report header name is “Patch Report”.
/PATCH_REPORT_SCORECARD/HEADER/GENERATION_DATETIME (#PCDATA)	The date and time when the report was generated.

XPath	element specifications / notes
/PATCH_REPORT_SCORECARD/HEADER/COMPANY_INFO	
	(NAME, ADDRESS, CITY, STATE, COUNTRY, ZIP_CODE)
	The user's company name and address, as defined in the user's account.
/PATCH_REPORT_SCORECARD/HEADER/USER_INFO	(NAME, USERNAME, ROLE)
/PATCH_REPORT_SCORECARD/HEADER/USER_INFO/NAME (#PCDATA)	
	The name of the user who generated the scorecard.
/PATCH_REPORT_SCORECARD/HEADER/USER_INFO/USERNAME (#PCDATA)	
	The user login ID of the user who generated the scorecard.
/PATCH_REPORT_SCORECARD/HEADER/USER_INFO/ROLE (#PCDATA)	
	The user role for the user who generated the scorecard: Manager, Unit Manager, Scanner or Reader.
/PATCH_REPORT_SCORECARD/SUMMARY (PARAM_LIST, DETAILS?)	
/PATCH_REPORT_SCORECARD/SUMMARY/PARAM_LIST	(PARAM+)
/PATCH_REPORT_SCORECARD/SUMMARY/PARAM_LIST/PARAM	(KEY, VALUE)
/PATCH_REPORT_SCORECARD/SUMMARY/PARAM_LIST/PARAM/KEY (#PCDATA)	
	A scorecard parameter name in the report source settings.
/PATCH_REPORT_SCORECARD/SUMMARY/PARAM_LIST/PARAM/VALUE (#PCDATA)	
	A scorecard parameter value in the report source settings.
/PATCH_REPORT_SCORECARD/SUMMARY/DETAILS (ASSET_GROUP_LIST)	
/PATCH_REPORT_SCORECARD/SUMMARY/DETAILS/ASSET_GROUP_LIST (ASSET_GROUP*)	
/PATCH_REPORT_SCORECARD/SUMMARY/DETAILS/ASSET_GROUP_LIST/ASSET_GROUP)	
	(TITLE, (STATS | DETECTION_LIST)
/PATCH_REPORT_SCORECARD/SUMMARY/DETAILS/ASSET_GROUP_LIST/ASSET_GROUP/ TITLE (#PCDATA)	
	An asset group title.
/PATCH_REPORT_SCORECARD/SUMMARY/DETAILS/ASSET_GROUP_LIST/ASSET_GROUP/STATS	
	(NUM_HOSTS?, SCANNED_HOSTS?, MISSING?)
/PATCH_REPORT_SCORECARD/SUMMARY/DETAILS/ASSET_GROUP_LIST/ASSET_GROUP/STATS/ NUM_HOSTS (#PCDATA)	
	The number of hosts in the asset group for which there is vulnerability scan data, followed in parentheses by the total number of IP addresses in the asset group.
/PATCH_REPORT_SCORECARD/SUMMARY/DETAILS/ASSET_GROUP_LIST/ASSET_GROUP/STATS/ SCANNED_HOSTS (#PCDATA)	
	The number of hosts in the asset group for which there is vulnerability scan data.
/PATCH_REPORT_SCORECARD/SUMMARY/DETAILS/MISSING	
	(ONE_OR_MORE_PATCHES?, SOFTWARE_1?, SOFTWARE_2?)
/PATCH_REPORT_SCORECARD/SUMMARY/ASSET_GROUP_LIST/ASSET_GROUP/STATS/DETAILS/ MISSING/ONE_OR_MORE_PATCHES	(PERCENT, TOTAL_HOSTS)

XPath	element specifications / notes
/PATCH_REPORT_SCORECARD/SUMMARY/ASSET_GROUP_LIST/ASSET_GROUP/STATS/DETAILS/MISSING/ONE_OR_MORE_PATCHES/ PERCENT (#PCDATA)	
	The percentage of scanned hosts in the asset group that are missing at least one of the user-specified patches.
/PATCH_REPORT_SCORECARD/SUMMARY/ASSET_GROUP_LIST/ASSET_GROUP/STATS/DETAILS/MISSING/ONE_OR_MORE_PATCHES/TOTAL_HOSTS (#PCDATA)	
	The number of scanned hosts in the asset group that are missing at least one of the user-specified patches.
/PATCH_REPORT_SCORECARD/SUMMARY/ASSET_GROUP_LIST/ASSET_GROUP/STATS/DETAILS/MISSING/SOFTWARE_1/ (PERCENT, TOTAL_HOSTS, QID?)	
/PATCH_REPORT_SCORECARD/SUMMARY/ASSET_GROUP_LIST/ASSET_GROUP/STATS/DETAILS/MISSING/SOFTWARE_1 /PERCENT (#PCDATA)	
	The percentage of scanned hosts in the asset group that are missing the first user-specified software QID.
/PATCH_REPORT_SCORECARD/SUMMARY/ASSET_GROUP_LIST/ASSET_GROUP/STATS/DETAILS/MISSING/SOFTWARE_1/TOTAL_HOSTS (#PCDATA)	
	The number of scanned hosts in the asset group that are missing the first user-specified software QID.
/PATCH_REPORT_SCORECARD/SUMMARY/ASSET_GROUP_LIST/ASSET_GROUP/STATS/DETAILS/MISSING/SOFTWARE_1/QID (#PCDATA)	
	The first user-specified software QID.
/PATCH_REPORT_SCORECARD/SUMMARY/ASSET_GROUP_LIST/ASSET_GROUP/STATS/DETAILS/MISSING/SOFTWARE_2 (PERCENT, TOTAL_HOSTS, QID?)	
/PATCH_REPORT_SCORECARD/SUMMARY/ASSET_GROUP_LIST/ASSET_GROUP/STATS/DETAILS/MISSING/SOFTWARE_2/ PERCENT (#PCDATA)	
	The percentage of scanned hosts in the asset group that are missing the second user-specified software QID.
/PATCH_REPORT_SCORECARD/SUMMARY/ASSET_GROUP_LIST/ASSET_GROUP/STATS/DETAILS/MISSING/SOFTWARE_2/TOTAL_HOSTS (#PCDATA)	
	The number of scanned hosts in the asset group that are missing the second user-specified software QID.
/PATCH_REPORT_SCORECARD/SUMMARY/ASSET_GROUP_LIST/ASSET_GROUP/STATS/DETAILS/MISSING/SOFTWARE_2/QID (#PCDATA)	
	The second user-specified software QID.
/PATCH_REPORT_SCORECARD/RESULTS (ASSET_GROUP_LIST)	
/PATCH_REPORT_SCORECARD/RESULTS/ASSET_GROUP_LIST/ASSET_GROUP_LIST (ASSET_GROUP*)	
/PATCH_REPORT_SCORECARD/RESULTS/ASSET_GROUP_LIST/ASSET_GROUP (TITLE, (STATS | DETECTION_LIST))	
/PATCH_REPORT_SCORECARD/RESULTS/ASSET_GROUP_LIST/ASSET_GROUP/TITLE (#PCDATA)	
	An asset group title.
/PATCH_REPORT_SCORECARD/RESULTS/ASSET_GROUP_LIST/ASSET_GROUP/STATS	
	(NUM_HOSTS?, SCANNED_HOSTS?, MISSING?)

XPath	element specifications / notes
/PATCH_REPORT_SCORECARD/RESULTS/ASSET_GROUP_LIST/ASSET_GROUP/STATS/NUM_HOSTS (#PCDATA)	
	The number of hosts in the asset group for which there is vulnerability scan data, followed in parentheses by the total number of IP addresses in the asset group.
/PATCH_REPORT_SCORECARD/RESULTS/ASSET_GROUP_LIST/ASSET_GROUP/SCANNED_HOSTS (#PCDATA)	
	The number of hosts in the asset group for which there is vulnerability scan data.
/PATCH_REPORT_SCORECARD/RESULTS/DETECTION_LIST (DETECTION*)	
/PATCH_REPORT_SCORECARD/RESULTS/DETECTION_LIST/DETECTION (HOST, VULN)	
/PATCH_REPORT_SCORECARD/RESULTS/DETECTION_LIST/DETECTION/HOST	
	(IP, DNS?, NETBIOS?, OS?, OWNER?)
/PATCH_REPORT_SCORECARD/RESULTS/DETECTION_LIST/DETECTION/HOST/IP (#PCDATA)	
	The IP address for a host missing required patches or software.
/PATCH_REPORT_SCORECARD/RESULTS/DETECTION_LIST/DETECTION/HOST/DNS (#PCDATA)	
	The registered DNS hostname for a host missing required patches or software.
/PATCH_REPORT_SCORECARD/RESULTS/DETECTION_LIST/DETECTION/HOST/NETBIOS (#PCDATA)	
	The NetBIOS hostname for a host missing required patches or software.
/PATCH_REPORT_SCORECARD/RESULTS/DETECTION_LIST/DETECTION/HOST/OS (#PCDATA)	
	The operating system detected on a host missing required patches or software.
/PATCH_REPORT_SCORECARD/RESULTS/DETECTION_LIST/DETECTION/HOST/OWNER (#PCDATA)	
	The owner of the host missing required patches or software.
/PATCH_REPORT_SCORECARD/RESULTS/DETECTION_LIST/DETECTION/VULN	
	(QID, VENDOR_REF?, TITLE)
/PATCH_REPORT_SCORECARD/RESULTS/DETECTION_LIST/DETECTION/VULN/QID	
	A vulnerability QID for a missing patch or software.
/PATCH_REPORT_SCORECARD/RESULTS/DETECTION_LIST/DETECTION/VULN/VENDOR_REF (#PCDATA)	
	A vendor reference for the vulnerability, such as a security bulletin.
/PATCH_REPORT_SCORECARD/RESULTS/DETECTION_LIST/DETECTION/VULN/TITLE (#PCDATA)	
	The title for the vulnerability for a missing patch or software.

APPENDIX F

Error Codes / Descriptions

Here's a list of Qualys API v2 error codes along with a description of what each code means. For an API request that had an error, you'll find the error code and text in the XML response.

HTTP Status	Error Code	Error Text	Meaning
HTTP/1.1 400 Bad Request	1901	Unrecognized parameter(s):...	The API request contained one or more parameters which are not supported, or are not available to the browsing user.
HTTP/1.1 400 Bad Request	1903	Missing required parameter(s):...	The API request did not contain one or more parameters which are required.
HTTP/1.1 400 Bad Request	1904	Please specify only one of these parameters:...	The API request contained 2 or more parameters from a group from which at most one may be specified.
HTTP/1.1 400 Bad Request	1905	parameter ... has invalid value ...	The API request contained a valid parameter specified with an invalid value.
HTTP/1.1 400 Bad Request	1907	The following combination of key=value pairs is not supported:...	The API request contained an invalid or unsupported combination of parameters.
HTTP/1.1 400 Bad Request	1908	Request method (GET or POST) is incompatible with specified parameter(s):...	The API request was made with an unsupported HTTP request method (GET or POST or PUT or DELETE or HEAD).

HTTP Status	Error Code	Error Text	Meaning
HTTP/1.1 409 Conflict	1920	The requested operation is blocked by one or more existing Business Objects	The API request was blocked by other API requests. In practice this should be replaced by one of error code 1960 or 1965 (see below).
HTTP/1.1 409 Conflict	1960	The requested operation is blocked by one or more existing Business Objects	Too many other API requests currently running (i.e. concurrency limit).
HTTP/1.1 409 Conflict	1965	The requested operation is blocked by one or more existing Business Objects	Too many other API requests have run recently (i.e. rate limit).
HTTP/1.1 400 Bad Request	1922	Please specify at least one of the following parameters:...	The API request was missing some required information (but not necessarily a single specific parameter).
HTTP/1.1 202 Accepted	1981	Your request is being processed. Please try this same request again later.	The API request is for a business operation which is already underway.
HTTP/1.1 400 Bad Request	999	Internal Error	The API request failed for some reason having to do with the (client) request. In practice this should always be expressed as some other error type, giving more information about what was actually wrong with the request.
HTTP/1.1 501 Internal Error	999	Internal Error	The API request failed due to a problem with QWEB.
HTTP/1.1 503 Maintenance	1999	We are performing scheduled maintenance on our System. We apologize for any inconvenience.	The API request failed because the Qualys Cloud Platform is in maintenance mode.
HTTP/1.1 401 Unauthorized	2000	Bad Login/Password	The API request failed because of an authentication failure.
HTTP/1.1 403 Forbidden	2002	User account is inactive.	The API request failed because of an authorization failure.
HTTP/1.1 409 Conflict	2003	Registration must be completed before API requests will be served for this account	The API request failed because nobody has yet accepted the EULA on behalf of the user's subscription.

HTTP Status	Error Code	Error Text	Meaning
HTTP/1.1 409 Conflict	2011	SecureID authentication is required for this account, so API access is blocked	The API request failed because SecureID authentication won't work with API calls.
HTTP/1.1 403 Forbidden	2012	User license is not authorized to run this API.	The API request failed because the user's subscription does not have API access enabled.

INDEX

A

API conventions 11
API limits 14
Appliance functions
    view scanner appliance list 67, 75
Asset functions
    purge hosts 168
    view host list 129, 136
    view IP list 126
asset group list output
    XPath elements 439
asset group vulnerability report 533
assign scanner appliance to network 353
authentication 10, 11, 253
authentication record list 257
authentication record list by type 260
authentication record list by type output
    XPath elements 511
authentication record list output
    XPath elements 507
authentication record return
    XPath elements 524
authentication using V2 APIs 18
authentication vault list output
    XPath elements 526
authentication vault view output
    XPath elements 529

B

batch return
    XPath elements 359
batch return DTD 171, 274, 282, 291, 296, 305, 337

C

characters in URLs 12
Cisco IOS authentication record 264
compliance authentication report
    XPath elements 488
compliance control list output
    XPath elements 445
Compliance functions
    view compliance control list 210
    view compliance policy list 216
    view compliance posture information 235
    view FDCC policy list 247
compliance policy export output
    XPath elements 456
compliance policy list output
    XPath elements 451, 501
compliance policy report
    XPath elements 477
compliance posture output
    XPath elements 464
compliance scan result output
    XPath elements 376
compliance scorecard report
    XPath elements 496
control criticality 241
Cyberscope report 241

D

date format 12
DTD for compliance authentication report 487, 492
DTD for compliance control list output 212, 444
DTD for compliance policy report 474
DTD for compliance scan result output 374
DTD for IPv6 mapping records list output 204
DTD for KnowledgeBase output 394
DTD for PCI scan share status output 381
DTD for report list output 407
DTD for scan list output 360, 363, 366
DTD for scanner appliance create output 392
DTD for scanner appliance list output 383

DTDs for assets
asset group list output 188, 437
excluded hosts change history output 428
excluded hosts list output 426
host list detection output 421
host list output 134, 416
IP list output 127, 174, 414
IPv6 mapping records list output 433
restricted IPs list output 435
scanner appliance list output 70, 87
virtual host list output 172, 431
DTDs for compliance policies
compliance policy export output 455
compliance policy list output 219, 449, 500
compliance posture information output 241, 461
FDCC policy list output 249
DTDs for reports
host list detection 151
report batch return 171
report list output 93
report simple return 40, 109, 117, 119, 124
DTDs for scan authentication
authentication record list by type output 261, 510
authentication record list output 258, 506
authentication record return 274, 282, 291, 296, 305, 337, 523
authentication vault list output 525
authentication vault view output 528
DTDs for scorecard reports
asset group vulnerability report 532
ignored vulnerabilities report 535
most prevalent vulnerabilities report 539
most vulnerable hosts report 543
patch report 547
DTDs, most recent 13

E

excluded hosts change history output
XPath elements 429
excluded hosts list output
XPath elements 427
Expires header 22

G

GET method 11

H

header parameter 18
host list detection
DTD 151
host list detection DTD 151
host list detection output
XPath elements 422
host list output
XPath elements 418
HTTP Expires header 22

I

ignored vulnerabilities report 536
IP list output
XPath elements 415
IPv4 to IPv6 asset mapping functions
remove IPv4 to IPv6 mapping records list 207
IPv6 asset mapping functions
add IPv6 mapping records 205
view IPv6 mapping records list 203

K

KnowledgeBase output
XPath elements 397

M

most prevalent vulnerabilities report 540
most vulnerable hosts report 544
MS SQL authentication record 331

N

Network functions
assign scanner appliance to network 353
create a network 351
network list 350
update a network 352

network list 350
XPath elements 410
network security audits 29
network support 349

O

Oracle authentication record 283
Oracle Listener authentication record 292

P

patch report 548
PCI scan share status output
XPath elements 382
POST method 11

Q

Qualys
API server 11
user account 10
vulnerability scans 29
Qualys API server 11
Qualys Support 7
Qualys user account 10

R

report batch return
DTD 171
report DTDs, most recent 13
report list output
DTD 93
XPath elements 408
Report Share functions
cancel report 118
delete report 123
download report 120
launch report 94
launch scorecard 110
report list 91
`report_template_list.php` 105
session login/logout 24, 26
report simple return
DTD 40, 109, 117, 119, 124
reports
batch return 171
Cyberscope report 241
date format 12
decoding reports 13
host list detection 151
report list output 93
simple return 40, 109, 117, 119, 124
restricted IPs list output
XPath elements 436

S

Scan Authentication functions
authentication record list 257
authentication record list by type 260
Cisco IOS authentication record 264
MS SQL authentication record 331
Oracle authentication record 283
Oracle Listener authentication record 292
SNMP authentication record 297
Unix authentication record 264
Windows authentication record 275
Scan functions
cancel running scan 30
KnowledgeBase 79
share PCI scan 83
scan list output
XPath elements 361, 364, 368
scanner appliance create output
XPath elements 393
scanner appliance list output
XPath elements 385
scans 29
session timeout 23
simple return
XPath elements 357
simple return DTD 40, 109, 117, 119, 124
SNMP authentication record 297
special characters in URLs 12

U

Unix authentication record 264
URL elements 12
URL encoded variables 12
user account
    login credentials 10
UTF-8 encoding 12

V

virtual host list output
    XPath elements 432, 434

W

Windows authentication record 275

X

XML output
    Asset 413
    authentication records 505, 531
    compliance data 443
    scan and report share 355
    scan authentication 505, 531